

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК 004.42

«До захисту допущено»
Завідувач кафедри
О.В. Коваль
(підпис) (ініціали, прізвище)
“ ” 2019р.

Магістерська дисертація

зі спеціальності - 122 Комп'ютерні науки

за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання
процесів і систем

на тему: «Система супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації»

Виконав: студент VI курсу, групи ТМ-81мп

Врадій Дмитро Вікторович
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент, к.т.н. Коваль О.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент к.т.н. Сенченко В. Р.
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Київ – 2019

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 122 Комп'ютерні науки

за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання процесів і систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

О.В. Коваль

(підпис)

” ____ ” _____ 2019р.

З А В Д А Н Н Я

НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ

Врадію Дмитру Вікторовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації: «Система супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації»

Науковий керівник Коваль Олександр Васильович, к.т.н., доцент

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджені наказом по університету від ”04” листопада 2019 р. № 3812-с

2. Строк подання студентом дисертації 9 грудня 2019р.

3. Вихідні дані до роботи магістерської дисертації

мова програмування TypeScript та Kotlin, середовище розробки JetBrains PhpStorm та Android Studio, фреймворк Angular

4. Перелік завдань, які потрібно розробити

проаналізувати проблему вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації, розробити систему супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації

5. Перелік ілюстративного матеріалу

схеми архітектури додатку, знімки інтерфейсу додатку та структури проекту

6. Орієнтований перелік публікацій

Міжнародна наукова інтернет-конференція на тему "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення" (випуск 44). Система

супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації

7. Дата видачі завдання "10" грудня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітки
1.	Затвердження теми роботи	10.12.18	
2.	Вивчення та аналіз задачі	11.02 – 11.03.19	
3.	Розробка архітектури та загальної структури системи	12.03 – 31.05.19	
4.	Розробка структур окремих підсистем	1.06 – 31.06.19	
5.	Програмна реалізація системи	03.09 – 22.10.19	
6.	Оформлення пояснювальної записки	01.11 – 07.12.19	
7.	Захист програмного продукту	23.10.19	
8.	Передзахист	20.11.19	
9.	Захист	16.12.19	

Студент _____
(підпис)

Науковий керівник _____
(підпис)

Врадій Д. В.
(прізвище та ініціали)

Коваль О. В.
(прізвище та ініціали)

РЕФЕРАТ

Магістерська дисертація складається зі вступу, п'яти розділів, висновку, переліку посилань з 21 найменування, одного додатку, і містить 40 рисунків, 20 таблиць. Повний обсяг магістерської дисертації складає 86 сторінок, з яких перелік посилань займає 2 сторінки, додаток – 5 сторінок.

Актуальність теми. З року в рік постає проблема покращення якості процесу вступу до вищих навчальних закладів. Було розроблено автоматизовану систему, яка спрощує процес подання вступних заяв та перегляду змін в рейтингових списках. Дана система покриває перший та другий освітні рівні вищої освіти, тому проблема автоматизації процесу для цієї категорії вступників вирішена. Але існує ще третій освітній рівень вищої освіти, для якого не існує системи автоматизації процесу вступу. Це зумовлює актуальність розробки нових програмних засобів тому, що автоматизація процесу має на меті економію ресурсів, пришвидшення подачі та перевірки вступних заяв, недопустимість припущення помилок людського фактору.

Мета дослідження полягає в розробці засобів, виявлення нових підходів проведення вступної компанії до вищого навчального закладу на третій освітній рівень та створення системи супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації.

Для досягнення поставленої задачі були сформульовані наступні **завдання дослідження**, що визначили логіку дослідження та його структуру:

- проаналізувати існуючі програмні рішення;
- визначити основні завдання для реалізації програмного продукту;
- проаналізувати процес вступу на третій освітній рівень;
- удосконалити існуючі процеси роботи з системою вступу;
- розробити програмний продукт для системи супроводження вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації.

Об'єктом дослідження є комп'ютерні інформаційні системи і технології.

Предметом дослідження є комп'ютерні інформаційні системи з використанням новітніх технологій розробки програмного забезпечення.

Наукова новизна одержаних результатів. Найбільш суттєвими науковими результатами магістерської дисертації є:

- удосконалено процес супроводження вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації та адміністрування документообороту під час вступної компанії;
- набуло подальшого розвитку використання автоматизованих систем в освітньому процесі.

Практичне значення одержаних результатів роботи полягає в розробці системи супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації.

Апробація результатів дисертації

Основні положення роботи доповідались і обговорювались на:

1. Міжнародній науковій інтернет-конференції на тему "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення" (випуск 44).

Ключові слова. *ВЕБ-СИСТЕМА, МОБІЛЬНИЙ ДОДАТОК, ХМАРНА ПЛАТФОРМА, КРОСПЛАТФОРМНІСТЬ, ВСТУП.*

ABSTRACT

Master's Thesis consists of an introduction, five sections, a conclusion, a list of references of 21 titles, one application, and contains 40 figures, 20 tables. The full volume of the master's thesis is 86 pages, of which the list of links occupies 2 pages, the application - 5 pages.

Actuality of theme. From year to year, there is the problem of improving the quality of the process of admission to higher education. An automated system has been developed to facilitate the application process and review of changes in the rating lists. This system covers the first and second levels of higher education, so the process automation problem for this category of entrants is solved. But there is still a third level of higher education for which there is no system for automation of the entry process. This determines the urgency of the development of new software tools because the automation of the process aims to save resources, accelerate the submission and verification of applications, the inadmissibility of human error.

The purpose of the study is to develop tools, identify new approaches to conducting admission to higher education at the third level of education and to create a system of support and analysis of entry of specialists of scientific and pedagogical staff of higher qualification.

To achieve this task, the following **research objectives** were formulated, which defined the logic of the study and its structure:

- analyze existing software solutions;
- define the main tasks for the implementation of the software product;
- analyze the process of entry to the third educational level;
- improve the existing processes of work with the entry system;
- develop a software product for the system of support of entry of specialists of scientific and pedagogical staff of higher qualification.

The object of study is computer information systems and technologies.

The subject of the study is computer information systems using the latest software development technologies.

Scientific novelty of the obtained results. The most significant scientific results of the master's thesis are:

- improved the process of supporting the admission of specialists of scientific and pedagogical staff of higher qualification and administration of document circulation during the introductory company;
- the use of automated systems in the educational process has been further developed.

The practical importance of the results obtained is to develop a system of support and analysis of entry of specialists of scientific and pedagogical staff of higher qualification.

Testing the results of the thesis

The main provisions of the work were reported and discussed at:

1. International Scientific Internet Conference on "Information Society: Technological, Economic and Technical Aspects of Becoming" (Issue 44).

Keywords. *WEB-SYSTEM, MOBILE APP, CLOUD PLATFORM, CROSS-PLATFORM, ADMISSION.*

ЗМІСТ

Перелік умовних позначень, символів, скорочень і термінів.....	10
Вступ	11
1 Задача розробки системи супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації.....	12
1.1 Характеристика системи	12
1.2 Висновки до розділу	17
2 Огляд методів реалізації програмної системи	18
2.1 Обрані методи реалізації.....	18
2.1.1 Angular 8	18
2.1.2 HTML.....	20
2.1.3 CSS	21
2.1.4 SASS.....	21
2.1.5 TypeScript	22
2.1.6 Kotlin	23
2.2 Висновки до розділу	24
3 Опис програмного забезпечення системи	26
3.1 Опис програмної реалізації веб-додатку	26
3.1.1 Модуль особистого кабінету адміністратора.....	27
3.1.2 Модуль реєстрації вступної заяви абітурієнта	29
3.1.3 Модуль перевірки рейтингового місця.....	30
3.1.4 Модуль ядра додатку.....	31
3.1.5 Спільний модуль.....	31
3.1.6 Observable	32
3.2 Опис програмної реалізації Android додатку.....	35
3.3 Опис програмного використання платформи Firebase	36
3.3.1 Створення документів колекції.....	37
3.3.2 Створення підколекцій.....	38

3.3.3 Виконання запитів	38
3.3.4 Видалення даних	38
3.4 Висновки до розділу	38
4 Інтерфейс та робота користувача з системою	40
4.1 Інтерфейс системи	40
4.1.1 Функціональні можливості абітурієнта	41
4.1.2 Функціональні можливості адміністратора	49
4.2 Висновки до розділу	53
5 Розроблення стартап-проекту	55
5.1 Опис ідеї проекту	55
5.2 Технологічний аудит ідеї проекту	64
5.3 Аналіз ринкових можливостей запуску стартап-проекту	65
5.4 Розроблення ринкової стратегії проекту	73
5.5 Розроблення маркетингової програми стартап-проекту	75
5.6 Висновки до розділу	78
Висновки	79
Список використаних джерел	80
Додаток А	82

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

ОС – операційна система

TypeScript – мова програмування базована на мові JavaScript

Kotlin – мова програмування для розробки на ОС Android

Android – мобільна операційна система

HTML – мова розмітки веб сторінки

CSS – каскадна таблиця стилів

PhpStorm – середовище розробки на мові програмування TypeScript

Android Studio – середовище розробки програмування на ОС Android

Фреймворк – програмна платформа, яка визначає структуру програмної системи

Маршрутизація – визначення маршрутів до додатку

ВСТУП

Розв'язком проблеми вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації, було розроблено систему прийому заяв на навчання, відображення рейтингової інформації вступника та адміністративною частиною, за допомогою якої можна аналізувати вхідні заяви та обробляти їх у порядку визначеному адміністратором системи.

Система призначена для супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої категорії. Система складається з двох частин, призначених для користувача та адміністратора.

Перша підсистема дозволяє вступникам подавати документи для вступу до аспірантури у електронному вигляді, відслідковувати прогрес вступу (залікові бали, рейтингові списки).

Друга підсистема дозволяє контролювати процес вступу адміністратором вступного процесу, приймати та редагувати подані заяви, вносити залікові бали, формувати рейтингові списки.

Система слугує спрощенням наявного процесу вступу, автоматизує ручну роботу та полегшує супроводження даних учасників вступної кампанії.

У першому розділі пояснювальної записки сформульовано мету і описано постановку задачі.

У другому розділі розроблений стартап-проект створеного програмного продукту.

У третьому розділі проведено огляд методів реалізації програмної системи.

У четвертому розділі описується програмна реалізація створеної системи.

В п'ятому розділі подається інструкція користувача для роботи з системою та наводиться її інтерфейс.

1 ЗАДАЧА РОЗРОБКИ СИСТЕМИ СУПРОВОДЖЕННЯ ТА АНАЛІЗУ ВСТУПУ СПЕЦІАЛІСТІВ НАУКОВО-ПЕДАГОГІЧНИХ КАДРІВ ВИЩОЇ КВАЛІФІКАЦІЇ

Метою розробки є реалізація програмного продукту на основі веб та мобільного додатків, що надає можливість супроводжувати вступну компанію спеціалістів науково-педагогічних кадрів вищої кваліфікації [1].

Призначенням даного програмного засобу є надання чіткого зрозумілого інтерфейсу для вступників та адміністраторів вступної компанії. При розробці програмного забезпечення розв'язані наступні завдання:

- а) проведено аналіз схожих систем на ринку;
- б) створене технічне завдання для системи;
- в) змодельована структура системи;
- г) розроблено програмне забезпечення системи;
- д) протестовано розроблену систему на функціональність.

Необхідними можливостями, які має забезпечувати система, є:

- а) доступність участі абітурієнта у вступній компанії;
- б) реалізація автоматичного розрахунку рейтингових списків вступників;
- в) супроводження вступної компанії з боку адміністратора;
- г) аналітика вступної компанії за різними критеріями;
- д) кросплатформність системи, що дозволяє використовувати додаток на великій кількості пристроїв.

1.1 Характеристика системи

Основними функціями програмного продукту з боку вступника є:

- а) функція подачі реєстраційної заяви для участі у вступній компанії;

б) функція відслідковування результатів вступної кампанії до аспірантури абітурієнтом;

в) перегляд результатів іспитів, рейтингового балу та позиції;

г) функціональність з точки зору адміністратора представлена такими можливостями;

д) функція перегляду всіх вступних заяв;

е) функція редагування вступної заяви та її статусу прийому;

ж) функція редагування вхідної інформації вступника;

з) функція створення списку вступників, де вступники згруповані за відповідною інформацією про факультет, кафедру та спеціальність;

и) функція, що дозволяє встановлювати кожному студенту бал за складений іспит у межах певного списку;

к) функція додаткових балів, передбачає введення додаткової інформації, яка може впливати на загальний рейтинговий бал вступника;

л) функція виведення результату всіх вступників певного списку, фільтрацію за рейтинговим балом;

м) функція перегляду статистики вступної кампанії за різними критеріями.

Зовнішній інтерфейс користувача передбачає головну сторінку, з якої за допомогою гіперпосилань можна перейти на сторінку подачі реєстраційної форми або сторінку відстеження рейтингового балу.

Сторінка реєстраційної форми передбачає багато полів заповнення та списків вибору, а саме:

а) прізвище;

б) ім'я;

в) по батькові;

г) стать;

д) дата народження;

е) спеціальність (обрати із списку);

ж) факультет (обрати із списку);

з) кафедра (обрати із списку);

- и) форма навчання (обрати із списку);
- к) форма оплати (обрати із списку);
- л) іноземна мова вступника (обрати із списку);
- м) ВНЗ, який закінчив вступник;
- н) рік випуску з ВНЗ;
- о) освітній ступінь (обрати із списку);
- п) освітній ступінь з відзнакою;
- р) електронна пошта;
- с) номер телефону;
- т) потрібність гуртожитка на час навчання;
- у) середній бал диплому за 100-бальною шкалою;
- ф) кількість публікацій;
- х) прізвище наукового керівника;
- ц) особливі відзнаки (переможець олімпіад, патенти, тощо);
- ч) додаткова інформація.

Після завершення реєстрації абітурієнтом вступної заяви, дані заносяться до бази даних і адміністратор системи може обробити реєстраційну форму вступника.

Зовнішній інтерфейс з точки зору адміністратора передбачає сторінку входу та розділи для роботи з даними вступників:

а) перегляд поданих заяв вступників, що відображаються у вигляді таблиці та формою пошуку вибраної заяви. Існує можливість перейти на сторінку детального відображення реєстраційної форми вступника;

б) детальна сторінка реєстраційної заяви, на якій відображаються дані внесені вступником на етапі реєстрації, а також кнопки підтвердження або відхилення в реєстрації;

в) список абітурієнтів вступної компанії;

г) детальна сторінка абітурієнта, з можливістю ввести бали за вступні екзамени та додаткові досягнення;

д) список зарахованих вступників;

е) сторінка перегляду статистики вступної компанії.

На рисунку 1.1 зображена діаграма прецедентів системи.

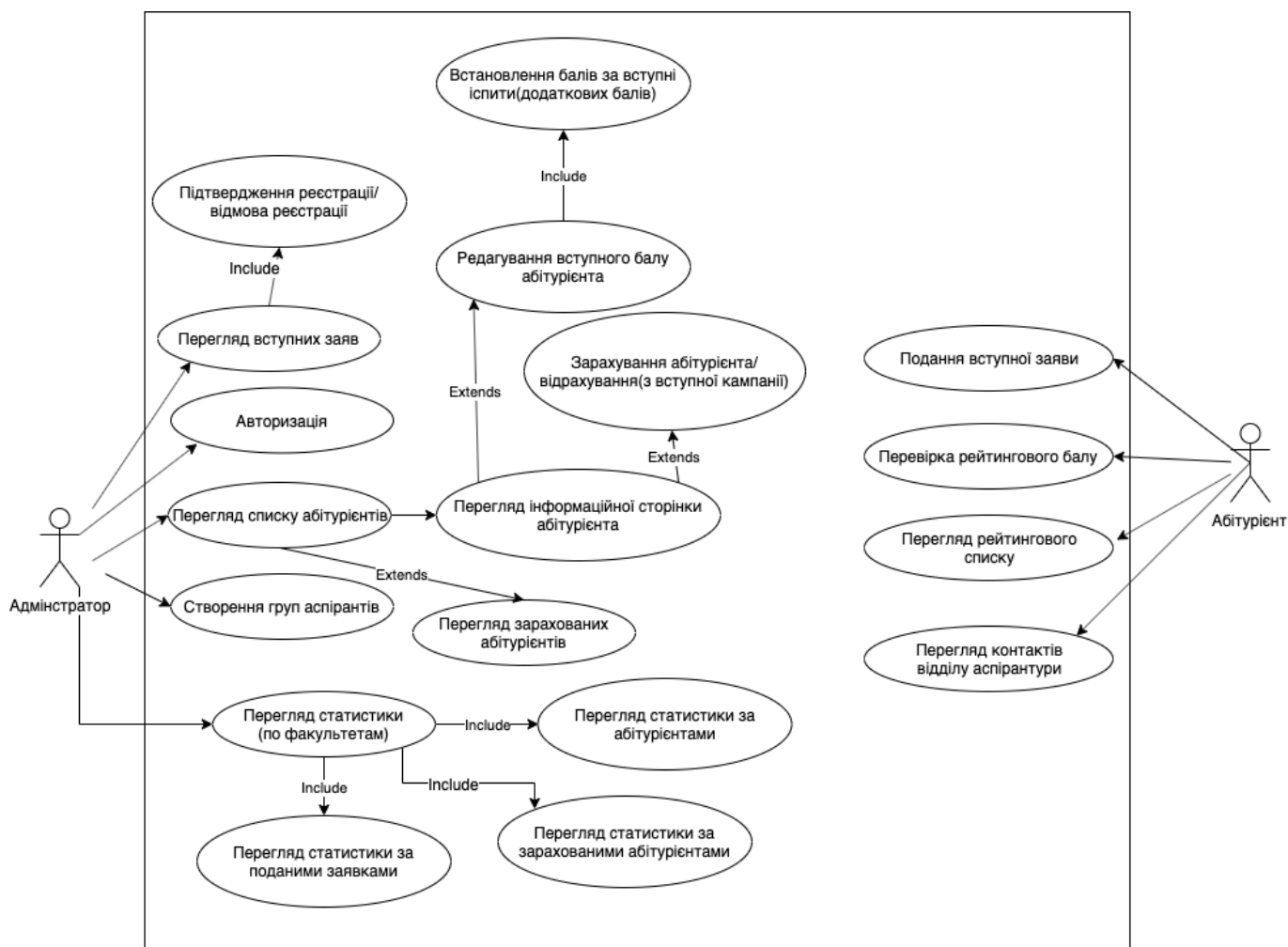


Рисунок 1.1 Діаграма прецедентів системи

На діаграмі послідовності системи для ролі абітурієнта (рисунок 1.2) зображена взаємодія користувача з елементами системи. Абітурієнт має змогу ввести свої дані до реєстраційної форми. Реєстраційна заява зберігається до бази даних системи і система повідомляє, що вступник успішно зареєстрований. Користувач має змогу перевірити свій рейтинговий бал, ввівши своє прізвище до пошукової форми. Результатом пошуку є рейтинговий список за даної спеціальністю.

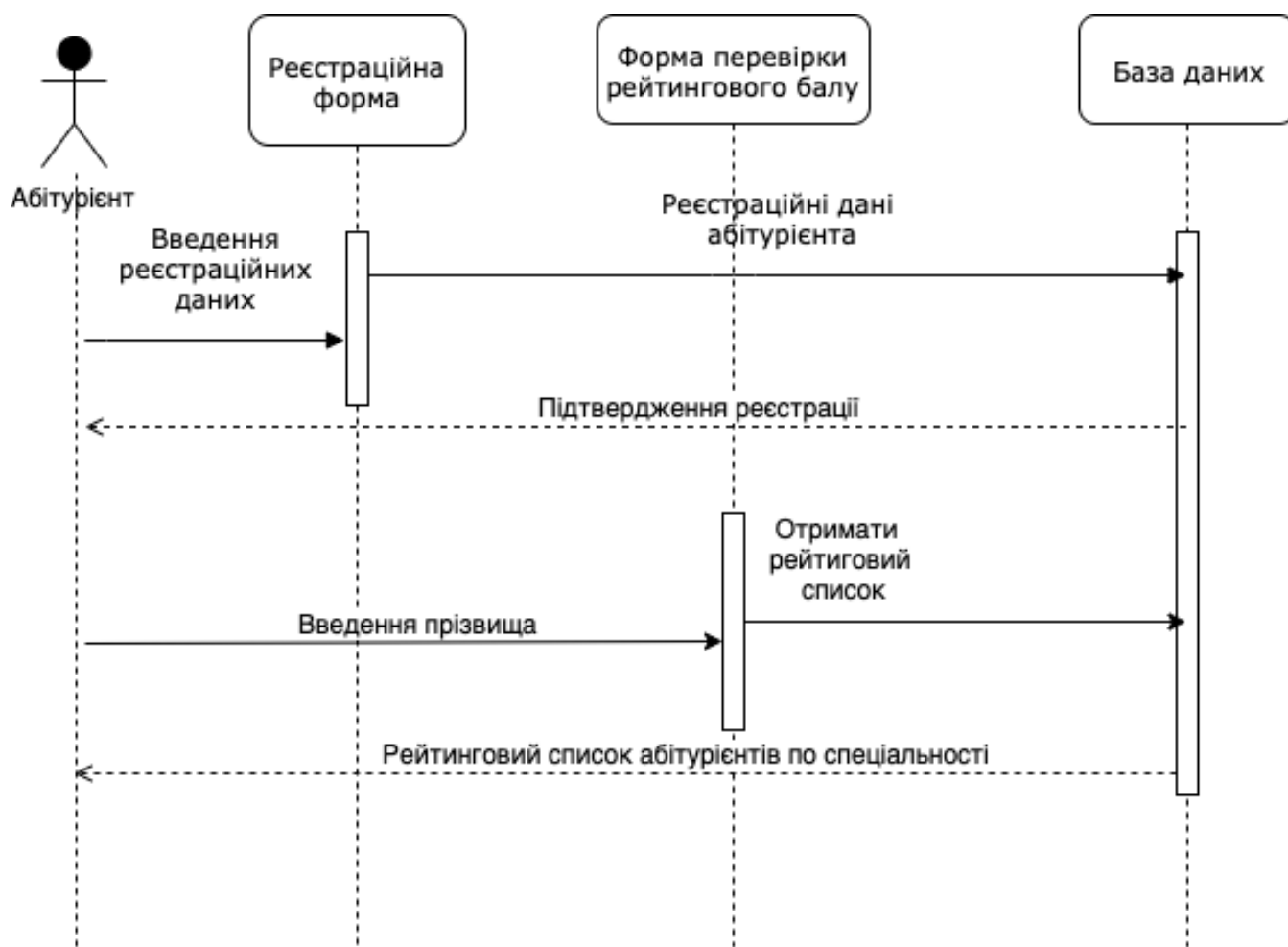


Рисунок 1.2 Діаграма послідовності системи. Роль абітурієнта

На діаграмі послідовності системи для ролі адміністратора (рисунок 1.3) зображена взаємодія користувача з елементами супроводження вступу спеціалістів науково-педагогічного спрямування. Адміністратор має змогу авторизуватись в системі за допомогою персонального логіну та паролю. Після успішного входу користувач має змогу переглядати реєстраційні заяви вступників, підтверджувати їх реєстрацію або відмовляти в ній, переглядати список підтверджених абітурієнтів, вносити бали за вступні екзамени та зберігати зміни. Наслідком збереження змін є формування рейтингового балу вступника та відображення даних у рейтинговій таблиці сформованій по спеціальності. Також адміністратор може зарахувати вступника до університету і присвоїти йому номер групи. Сторінка статистики дозволяє побачити графічну ілюстрацію аналітики вступної компанії.

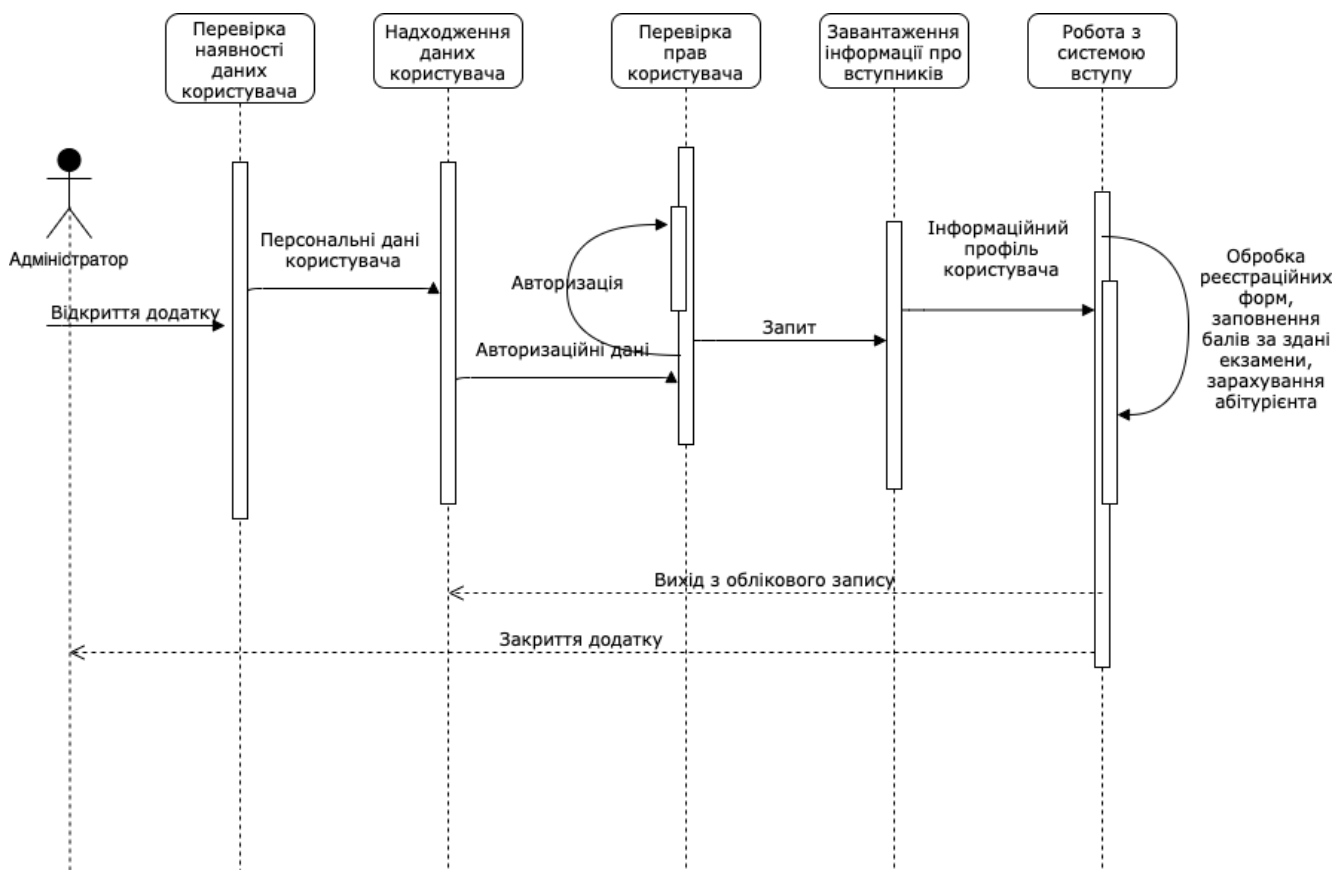


Рисунок 1.3 Діаграма послідовності системи. Роль адміністратора

1.2 Висновки до розділу

У цьому розділі наведено завдання вирішення проблеми супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації.

Система може бути поділена на дві функціональні підсистеми:

- підсистема абітурієнта;
- підсистема адміністратора.

Абітурієнт має змогу подавати реєстраційні заяви та переглядати рейтингові списки вступників.

Адміністратор приймає рішення у реєстрації вступника в системі, а також супроводженням картки абітурієнта після проходження вступних екзаменів. Адміністратору доступна сторінка перегляду статистики вступної компанії.

2 ОГЛЯД МЕТОДІВ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

Під час розробки програмного продукту є правильний вибір засобів програмної реалізації та технологій. Це дає змогу вплинути на час розробки, якість та надійність.

Розробка системи проводилась у середовищі JetBrains PhpStorm 2019, як засіб для роботи з веб-розробкою та Android Studio, як засіб для роботи з програмуванням під ОС Android.

2.1 Обрані методи реалізації

Для реалізації веб-системи було використано мову програмування TypeScript та фреймворк Angular 8, для мобільного додатку було використано мову програмування Kotlin. Поєднання цих технологій дає змогу створити кросплатформний додаток. Це означає, що даний застосунок можна використовувати на великій кількості платформ: Android, Windows, Linux, MacOS, та ін.

Основними вимогами до програмного забезпечення, є наявність операційної системи, яка має встановлений браузер для перегляду сайтів або мобільний пристрій на ОС Android. Перевагою системи є те, що мобільні пристрої мають змогу використовувати систему. Для цього потрібно скомпілювати додаток та встановити на мобільний пристрій.

2.1.1 Angular 8

Angular – це написаний на TypeScript веб-фреймворк, який розроблений в компанії Google. Перш за все він направлений на розробку односторінкових веб-додатків [5-7, 10, 13-14].

Angular 8 надає таку функціональність як:

- а) двостороннє зв'язування;
- б) використання шаблонів;

- в) маршрутизація;
- г) динамічне завантаження модулів;
- д) впровадження залежності за допомогою сервісів.

Можна досягнути максимально можливої швидкості на веб-платформі, а потім взяти її за допомогою Web Workers та рендеринга на стороні сервера.

Angular дозволяє контролювати масштабованість. Можна задовольняти величезні вимоги до даних, створюючи моделі даних на RxJS, Immutable.js або іншу push-модель.

Існує можливість швидкого створення функції за допомогою простих, декларативних шаблонів. Розширення мови шаблонів проходить своїми власними компонентами та використовується широкий спектр існуючих компонентів. Можна отримати негайну спеціальну допомогу та відгуки щодо практичного використання з кожним IDE та редактором. Все це відбувається разом, щоб можна було зосередитися на створенні додатків.

Основними будівельними блоками Angular додатку є NgModules, які надають контекст компіляції для компонентів. NgModules збирають відповідний код у функціональні набори; Angular додаток визначається набором NgModules. Додаток завжди має принаймні кореневий модуль, який дозволяє завантажувальний файл, і зазвичай має багато інших функціональних модулів.

Компоненти визначають перегляди, які є наборами елементів екрану, які Angular може вибрати серед і змінювати відповідно до вашої логіки програми та даних. Кожний додаток має принаймні один кореневий компонент.

Компоненти користуються сервісами, які забезпечують особливі функції, не пов'язані безпосередньо з представленнями. Постачальники послуг можуть бути введені в компоненти як залежності, що робить код модульним, багаторазовим і ефективним (рисунок 2.1).

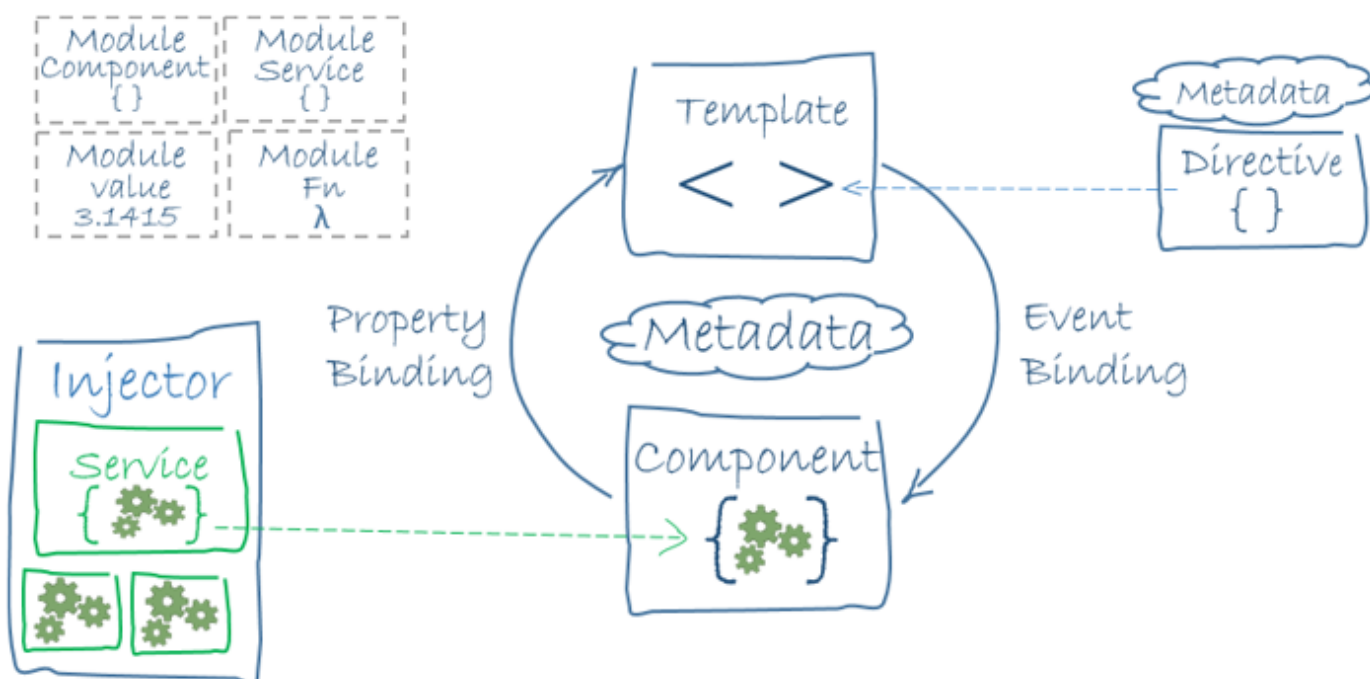


Рисунок 2.1 Схеми архітектури Angular додатка

Для роботи з Angular використовується інтерфейс командного рядка Angular CLI. З його допомогою можна використовувати такі команди:

- е) `ng new` – створює новий проект;
- ж) `ng generate` – генерує новий компонент або сервіс;
- з) `ng serve` – компілює додаток та вивантажує на локальний сервер;
- и) `ng test` – компілює додаток та проводить тест написаного коду;
- к) `ng build` – компілює та мінімізує проект в готовий додаток.

Щоб створити шаблон компонента була використана мова розмітки веб сторінок HTML та мова каскадних стилів CSS з препроцесором SASS.

2.1.2 HTML

HTML – це мова гіпертекстової розмітки, яка є стандартною мовою для створення веб сторінок. Веб-браузери отримують HTML-документи з веб-серверу або із локального сховища. HTML описує структуру веб-сторінки семантично, тобто кожен блок має свою сутність.

HTML-елементи є засобами для створення структурованих документів, описуючи структурну семантику для тексту, наприклад заголовки, абзаци, списки,

посилання, цитати та інші елементи. HTML-елементи позначаються тегами, написаними за допомогою кутових дужок.

HTML5 – це остання і основна версія стандарту HTML. Версія була створена, щоб покращити підтримку нових мультимедіа, одночасно зберігаючи його легко розуміючим людиною, так і пристроями такими як браузері, синтаксичні аналізатори та інші.

2.1.3 CSS

CSS - це мова для опису презентації веб-сторінок, включаючи кольори, макет і шрифти. Це дозволяє адаптувати презентацію до різних типів пристроїв, таких як великі екрани, маленькі екрани або принтери. CSS не залежить від HTML і може використовуватися з будь-якою мовою розмітки на основі XML. Відокремлення HTML від CSS полегшує підтримку сайтів, обмін таблицями стилів на сторінках та індивідуальність сторінок у різних середовищах. Це називається поділом структури (або: змісту) на презентацію.

Веб-браузері застосовують правила CSS до документа, щоб впливати на їх показ. Правило CSS формується з:

- а) набір властивостей, які мають значення для оновлення вигляду HTML-вмісту, наприклад, я хочу, щоб ширина мого елемента становила 50% від його батьківського елемента, а його фон відображався червоним;
- б) селектор, який вибирає елементи, якими ви хочете застосувати оновлені значення властивостей до. Наприклад, я хочу застосувати своє правило CSS до всіх абзаців в моєму документі HTML;

2.1.4 SASS

SASS (Syntactically awesome style sheets) – це мова сценаріїв препроцесора, який інтерпретується або компілюється в CSS, призначений для збільшення рівня абстракції CSS коду та спрощення файлів CSS [4].

Мова SASS має два синтаксису:

- а) sass – не використовує фігурні дужки, вкладені елементи реалізовані за допомогою табуляції;
- б) SCSS (Sassy CSS) – використовує фігурні дужки, як і сам CSS.

SASS додає до CSS константи і домішки. Це полегшує підтримку цілісності даних всередині великого набору стилів. Константи дозволяють встановити значення і використовувати його всередині стилів, за допомогою домішок те ж саме можна зробити з блоком атрибутів стилю.

Аналогами SASS являються препроцесори LESS та Stylus.

2.1.5 TypeScript

TypeScript – це мова програмування для розробки веб-додатків, яка збільшує можливості JavaScript (рисунк 2.2). Найбільшою перевагою над JavaScript [2-3] є вбудована підтримка типів. Після компіляції програма на TypeScript перетворюється в програму на JavaScript, яку можна використовувати в браузері.

В TypeScript можна використовувати наявний JavaScript код, додавати популярні бібліотеки JavaScript та викликати TypeScript код, згенерований з другого JavaScript. Об'явлення типів для цих бібліотек постачаються разом з вихідним кодом.

TypeScript забезпечує оголошення типів для статичної перевірки їх узгодження. Це не є обов'язковим і може бути проігноровано, щоб використовувати звичайну динамічну типізацію JavaScript [12, 15].

Щоб програми були корисними, ми повинні мати можливість працювати з деякими найпростішими одиницями даних: цифрами, рядками, структурами, логічними значеннями тощо. У TypeScript існує підтримка тих типів, що очікуються в JavaScript, з зручним набором переліку.

З введенням класів в TypeScript та ES6 існують певні сценарії, які потребують додаткових функцій для підтримки анотування чи модифікації класів та членів класу. Декоратори забезпечують спосіб додавання як анотацій, так і синтаксису мета-програмування для декларацій класів та учасників. Декоратори є пропозицією 2-го етапу для JavaScript і доступні як експериментальна функція TypeScript.

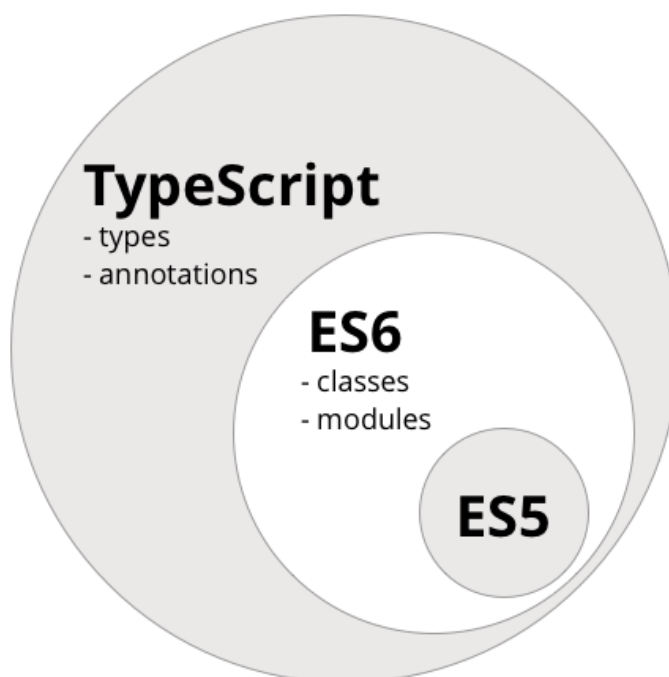


Рисунок 2.2 Розширення JavaScript за допомогою TypeScript

Особливості мови:

- а) анотація типів та їх перевірка на стадії компілювання;
- б) вивід типів;
- в) класи;
- г) інтерфейси;
- д) перелічувані типи даних;
- е) модулі;
- ж) анонімні функції;
- з) виклик асинхронного коду за допомогою await;

Ці нововведення підвищують швидкість розробки, рефакторинг, прочитання коду та його повторне використання.

2.1.6 Kotlin

Kotlin - це статично типова мова програмування загального призначення, розроблена JetBrains, яка створила IDE світового класу, такі як IntelliJ IDEA, PhpStorm, Appcode тощо. Вперше була введена JetBrains у 2011 році та новою мовою для JVM. Kotlin – об'єктно-орієнтована мова краща, ніж Java, але все-таки повністю взаємодіє з кодом Java [11, 16-20].

Kotlin спонсорується компанією Google, яка оголошена однією з офіційних мов розробки Android у 2017 році.

Основні особливості мови Kotlin:

- а) статично-типізована. Це мова програмування, яка означає тип кожної змінної і вираз відомий під час компіляції. Хоча це статично набрана мова, вона не вимагає чітко вказувати тип кожної змінної, яка оголошується;
- б) класи даних;
- в) лаконічність. Це різко зменшує зайвий код, написаний іншими об'єктно-орієнтованими мовами програмування;
- г) безпечність. Забезпечується безпека від більшості дратівливих `NullPointerException`, підтримуючи зведеність на нуль як частину своєї системи;
- д) сумісність з Java. Kotlin працює на віртуальній машині Java (JVM), тому вона повністю взаємодіє з Java;
- е) функціональні та об'єктно-орієнтовані можливості. Kotlin має великий набір багатьох корисних методів, що включають функції вищого порядку, лямбда-вирази, перевантаження оператора та багато іншого;
- ж) Smart Cast. Чітко типізуються незмінні значення та автоматично вставляється значення в безпечний формат;
- з) час компіляції. Має більш високу продуктивність і швидкий час компіляції;
- и) зручність для інструментів розробки.

2.2 Висновки до розділу

Для створення програмної системи було обрано середовище розробки JetBrains PhpStorm та Android Studio 2019, яке надає змогу працювати з веб та мобільною розробкою. TypeScript стала основною мовою програмування для веб розробки, а Kotlin – для розробки на операційній системі Android. HTML та SASS є засобами розмітки гіпертексту та стилізації.

Був обраний фреймворк Angular для створення веб додатку, який дозволяє створити швидку та потужну платформу і має широку підтримку в спільноті розробників.

Об'єднання технологій дає змогу створити швидкий та зручний кросплатформний додаток, який може створити конкуренцію для аналогів.

.

3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

Розроблена система становить собою веб та Android додаток, що має структуру представлену на рисунку 3.1.

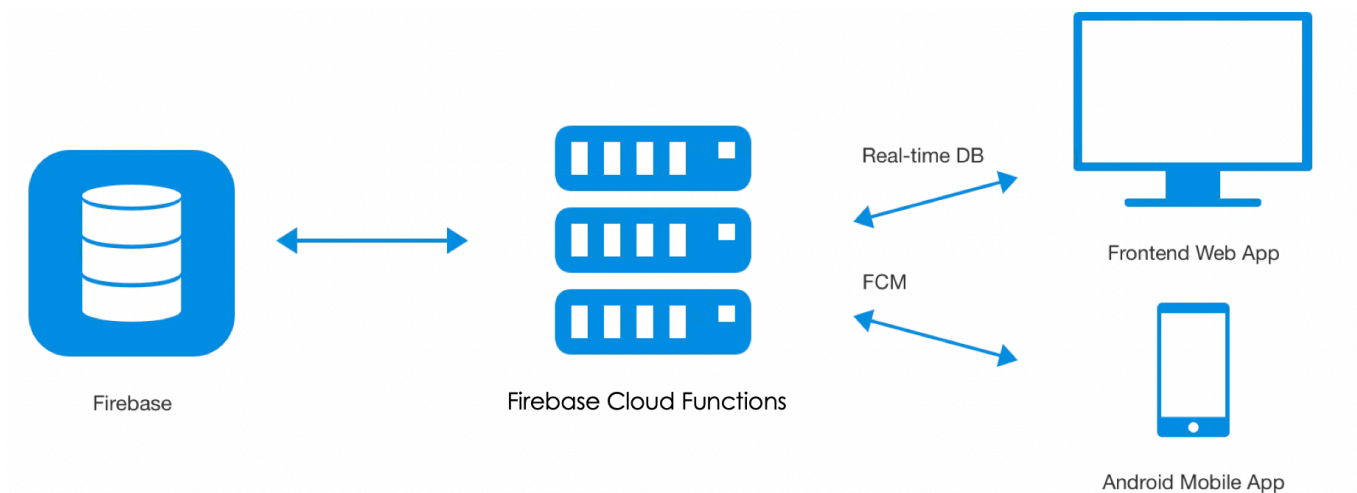


Рисунок 3.1 Структура системи

3.1 Опис програмної реалізації веб-додатку

Web-додаток створений за допомогою мови TypeScript та фреймворку Angular 8, що дає змогу створити швидко-зростаючий додаток має модульну структуру (рисунок 3.2) і складається з декількох модулів:

- а) модуль особистого кабінету адміністратора;
- б) модуль реєстрації вступної заяви абітурієнта;
- в) модуль перевірки рейтингового місця;
- г) модуль ядра додатку;
- д) спільний модуль.

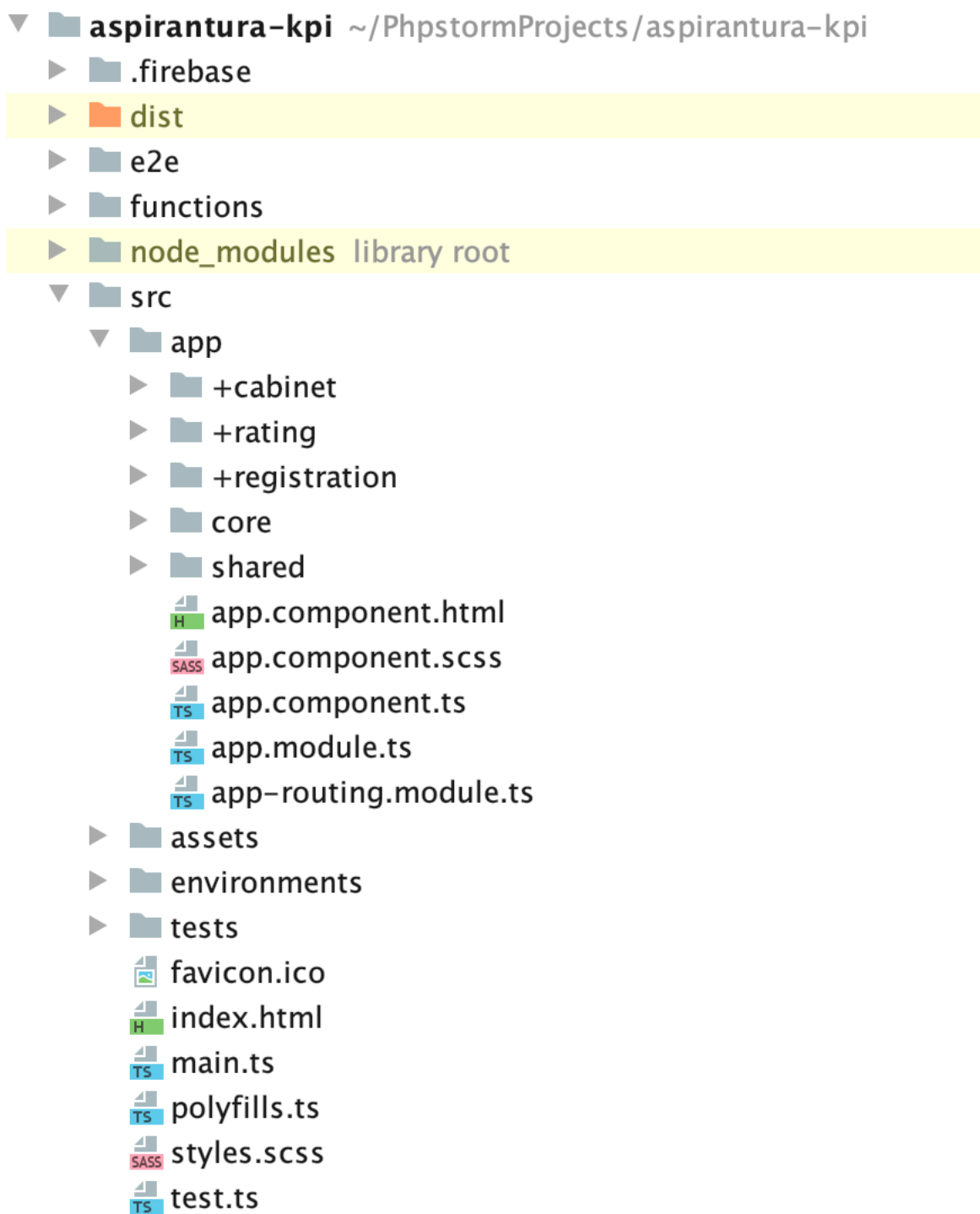


Рисунок 3.2 Структура проекту web-додатку

3.1.1 Модуль особистого кабінету адміністратора

Модуль особистого кабінету адміністратора складається з компонентів та сервісу, який взаємодіє з хмарним середовищем Firebase (рисунок 3.3).

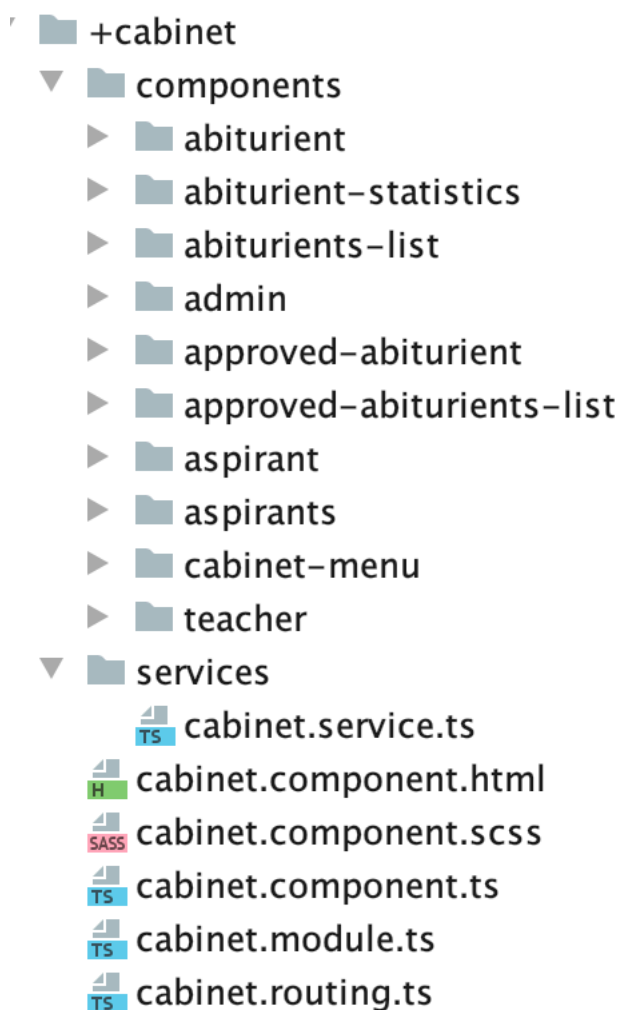


Рисунок 3.3 Структура модулю особистого кабінету адміністратора

Сервіс модулю особистого кабінету адміністратора за допомогою шаблону проектування впровадження залежності в конструктор використовує зовнішній сервіс Firebase та HttpClient для зв'язку з зовнішніми джерелами даних (рисунок 3.4).

```
@Injectable()
export class CabinetService {
  constructor(
    private _afs: AngularFirestore,
    private _authService: AuthService,
    private _httpClient: HttpClient,
  ) {}
```

Рисунок 3.4 Впровадження залежності в сервіс особистого кабінету адміністратора

За допомогою бібліотеки реактивного програмування RxJS, створені методи, що завантажують інформацію з баз даних Firebase в режимі реального часу. Тобто будь-які зміни в базі даних відразу будуть відображені в клієнтському застосунку. Для цього використовується шаблон проектування «Спостерігач». Створюється потік даних типу Observable, на який потрібно підписатись в компоненті (рисунок 3.5).

```
public getUser(): Observable<any> {
  return from(
    this._afs.collection<any>( path: 'users-info')
      .doc(this._authService.getUserId())
      .get()
      .pipe(map( project: (user: any) => {
        return {
          ...user.data(),
          id: user.id,
        };
      })));
}
```

Рисунок 3.5 Підписка на дані користувача

3.1.2 Модуль реєстрації вступної заяви абітурієнта

Модуль реєстрації вступної заяви абітурієнта складається з компонента та сервісу (рисунок 3.6). Сервіс RegistrationService має метод createStudent(), що створює реєстраційну заяву в базі даних.

Компонент подачі реєстраційної заяви створений за допомогою елементів Angular Material SDK. Це дозволяє використовувати компоненти в єдиному стилі та які цілком задовольняють користувацькі вимоги:

- а) matInput – рядок вводу строкових даних;
- б) matSelect – випадаючий елемент форми;
- в) matCheckbox – компонент, який приймає булеві значення;
- г) matLabel – помилка, яка відображається у формі;
- д) matLabel – підпис до рядку форми.

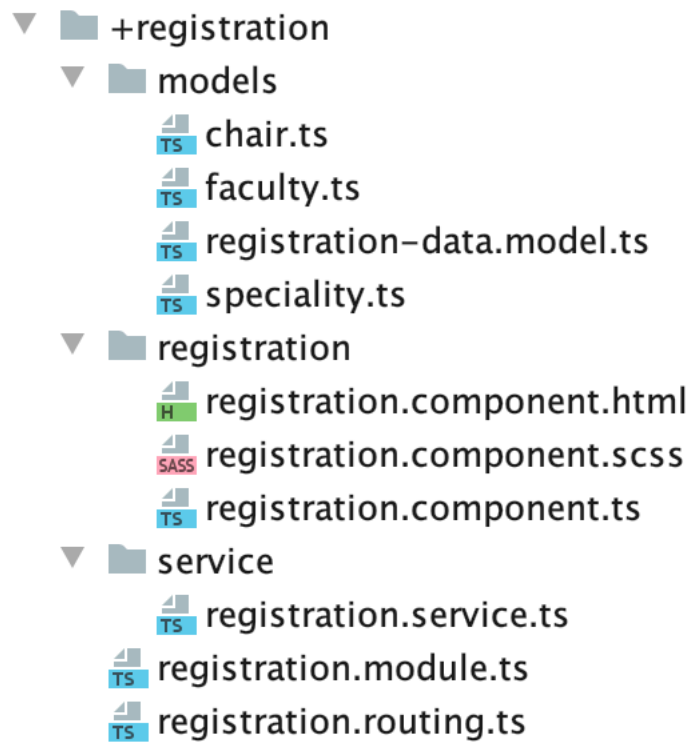


Рисунок 3.6 Структура модулю реєстрації вступної заяви абітурієнта

3.1.3 Модуль перевірки рейтингового місця

Модуль складається з двох компонентів та сервісу (рисунок 3.7). Сервіс дає змогу здійснити пошук абітурієнта та отримати рейтинговий список.

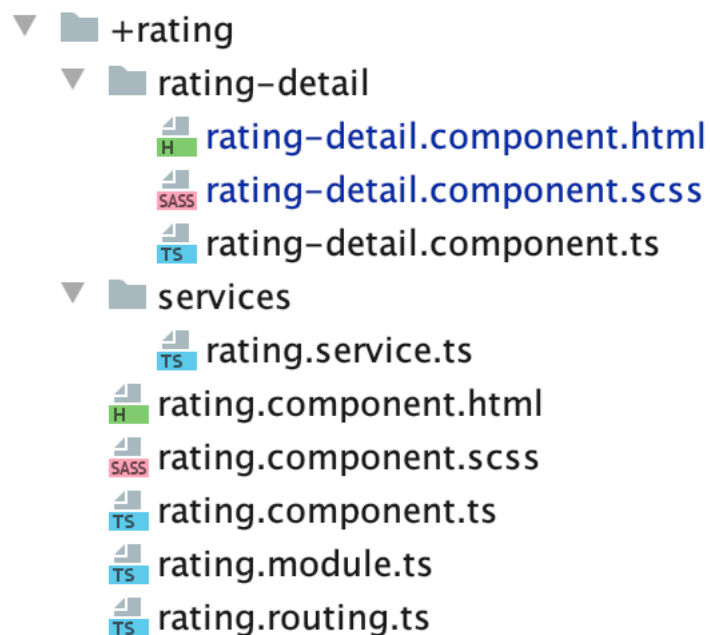


Рисунок 3.7 Структура модулю перевірки рейтингового місця

3.1.4 Модуль ядра додатку

Модуль ядра додатку складається з сервісу автентифікації та компонентів, які є загальними для всього додатку (рисунок 3.8).



Рисунок 3.8 Структура модулю ядра додатку

3.1.5 Спільний модуль

Спільний модуль імпортується в інші модулі і містить декларації компонентів Angular Material SDK (рисунок 3.9).

```

@NgModule({
  exports: [
    MatButtonModule, MatIconModule, MatFormFieldModule,
    MatInputModule, MatStepperModule, MatCardModule,
    MatSelectModule, MatCheckboxModule, MatSnackBarModule,
    MatDatepickerModule, MatNativeDateModule, MatSidenavModule,
    MatTableModule, MatDividerModule, MatListModule,
  ]
})
export class MaterialModule {}
  
```

Рисунок 3.9 Декларації компонентів Angular Material SDK

3.1.6 Observable

У реактивному програмуванні використовується термін Observable. Це лінійні push-колекції декількох значень. Спостерігач підписується на Observable і потім реагує на будь-який предмет або послідовність предметів, що продукує Observable (рисунок 3.10).

Ця закономірність полегшує одночасні операції, оскільки її не потрібно блокувати під час очікування, коли Observable продукує об'єкти, а натомість створює спостерігача у формі Observable, який готовий реагувати належним чином у будь-який майбутній час, коли це спостерігатиме.

Перевага такого підходу полягає в тому, що коли є маса завдань, які не залежать одне від одного, можна розпочати їх всі одночасно, а не чекати, коли кожне закінчиться, перш ніж розпочати наступне. Таким чином, весь пакет завдань займає лише стільки часу, скільки має найдовше завдання в пакеті.

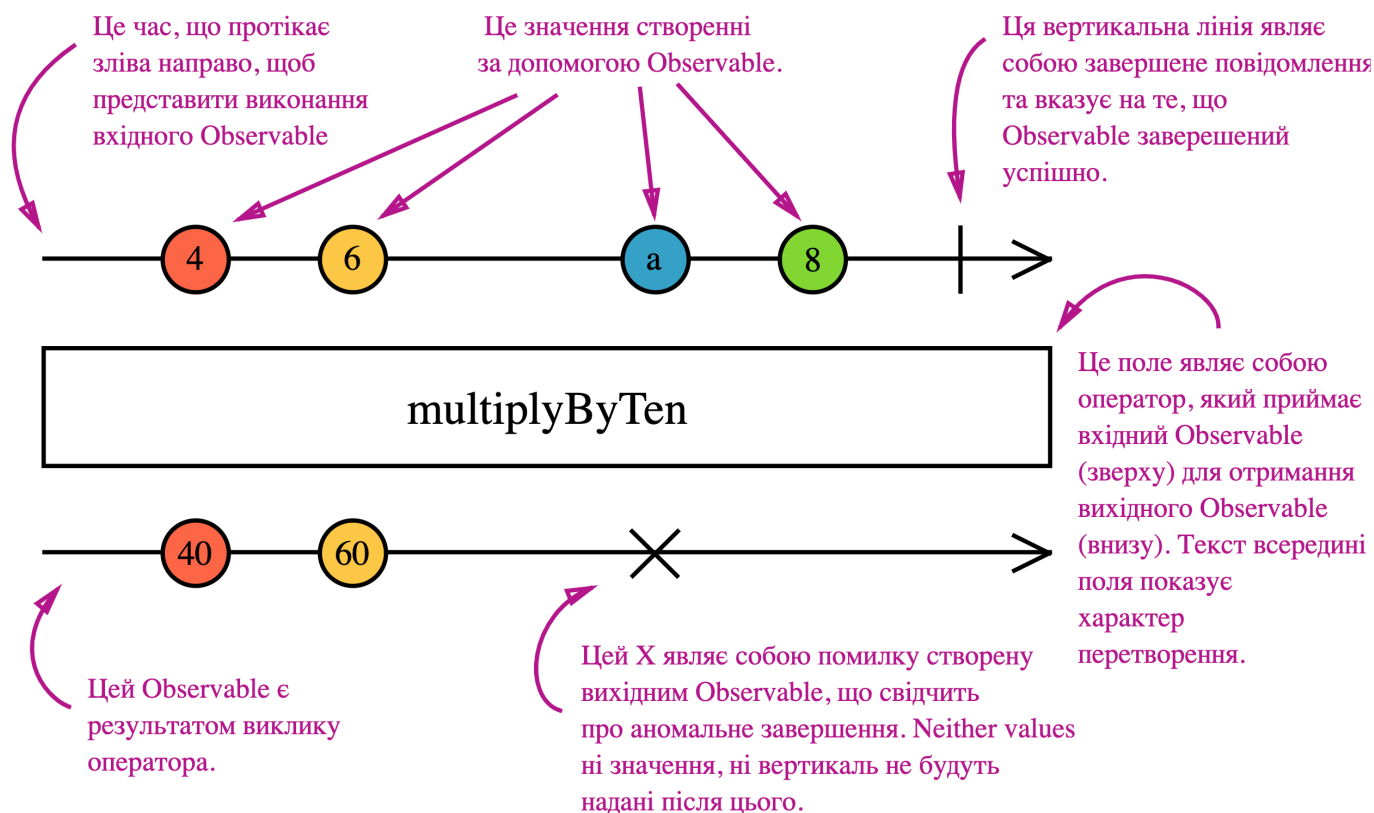


Рисунок 3.10 Представлення Observable у виді marble діаграми

Оператор `from` перетворює масив, об'єкт подібний на масив або `Promise` на тип `Observable`. На зображеній marble діаграмі (рисунок 3.11) схематично показано принцип роботи оператора.

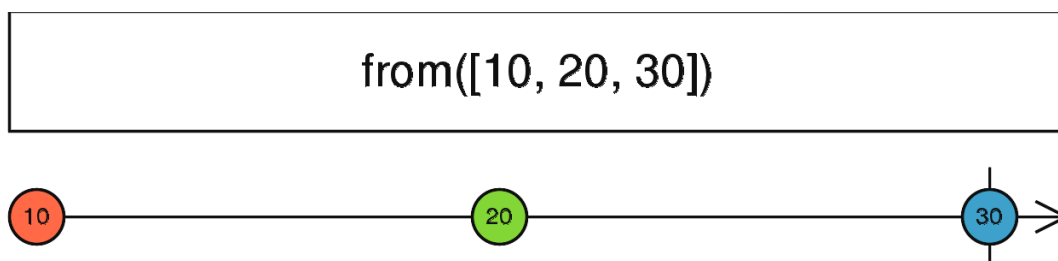


Рисунок 3.11 Marble діаграма оператора `from`

Оператор `map` застосовує задану функцію до кожного значення, що надходить від джерела `Observable` і повертає отримані значення як `Observable` (рисунок 3.12).

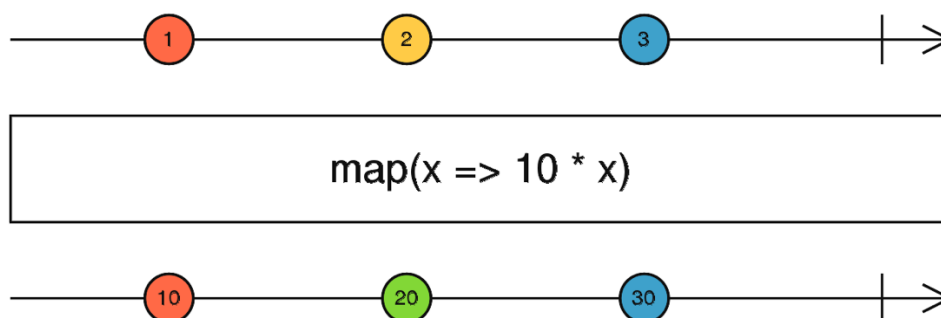
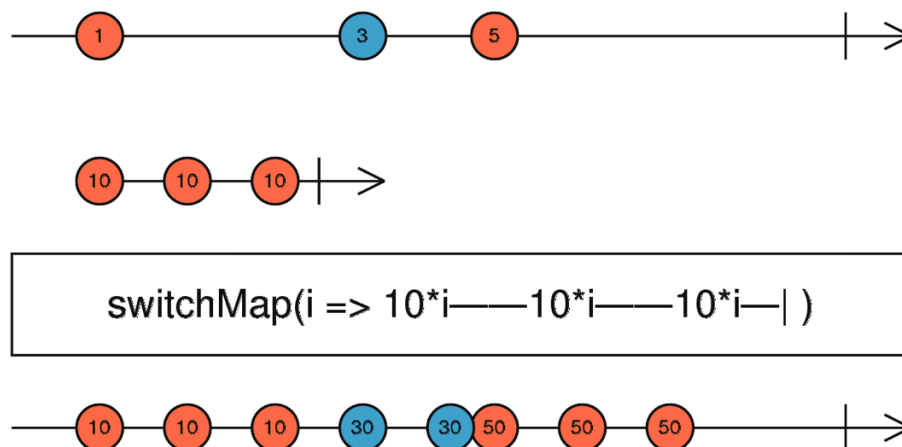


Рисунок 3.12 Marble діаграма оператора `map`

Важливим оператором є `switchMap` (рисунок 3.13), що повертає `Observable`, який публікує елементи на основі функції, яка застосовується до кожного елемента, створеного джерелом `Observable`, де ця функція повертає (так званий "внутрішній") `Observable`. Кожен раз, коли оператор спостерігає одну з цих внутрішніх `Observable`, вихідний `Observable` починає публікувати елементи, створені цим внутрішнім `Observable`. Коли створюється новий внутрішній `Observable`, `switchMap` припиняє створювати предмети із раніше створеного внутрішнього `Observable` і починає створювати елементи з нового. Він продовжує вести себе таким чином для наступних внутрішніх `Observable`.

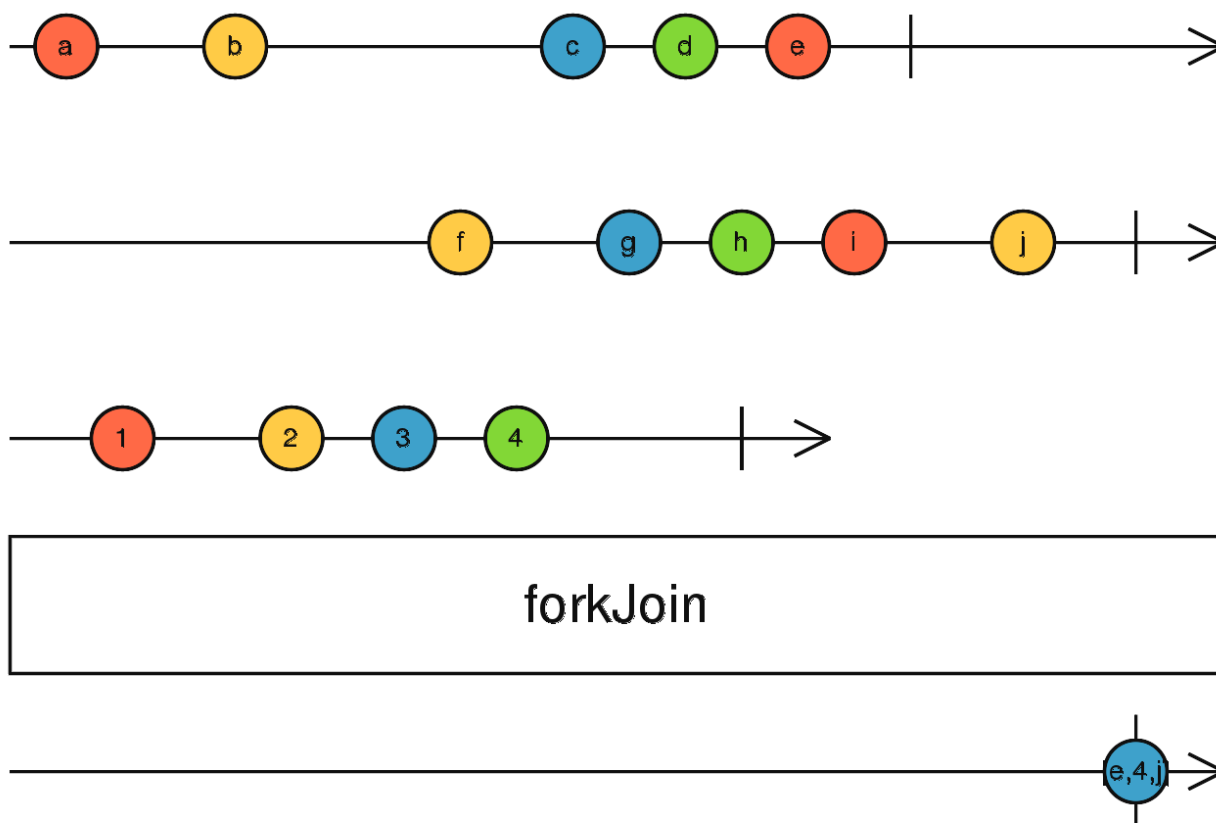
Рисунок 3.13 Marble діаграма оператора `switchMap`

Оператор `forkJoin` приймає будь-яку кількість спостережуваних даних, які можна передавати як масив, так і словник спостережуваних даних. Якщо спостереження за входом не надано, отриманий потік негайно завершиться (рисунок 3.14).

Оператор `forkJoin` буде чекати завершення всіх переданих спостережень, і тоді він буде випромінювати масив або об'єкт з останніми значеннями з відповідних спостережних даних.

Якщо передати оператору масив з n -видимих даних, в результаті масив матиме n -значень, де перше значення - це останнє, що випромінюється першим спостережуваним, друге значення - це останнє, що випускається другим спостережуваним тощо.

Якщо передати оператору словник спостережуваних даних, в результаті об'єкти матимуть ті самі ключі, що і словник, передані з останніми значеннями, які вони випустили, розташованими у відповідному ключі.

Рисунок 3.14 Marble діаграма оператора `forkJoin`

3.2 Опис програмної реалізації Android додатку

Для запуску і тестування програми можна використовувати емулятори або реальні пристрої. Але в ідеалі краще тестувати на реальних пристроях.

Для використання мобільного пристрою для тестування на робочу машину необхідно встановити драйвер. Якщо смартфон від Google - Nexus 5/6 / 5x / 6P або Google Pixel, то для його підтримки необхідно через SDK Manager встановити пакет Google Usb Driver. Якщо ж виробник апарату - інший вендор, то треба встановити той USB-драйвер, який поставляється даним вендором. Якщо ОС – Windows 10, то там, як правило, система сама може знайти через центр оновлень драйвер і встановити його.

За замовчуванням опції розробника на смартфонах приховані. Щоб зробити їх доступними, треба зайти в Settings> About phone (Установки> Про телефон) і сім раз натиснути Build Number (Номер збірки).

Розглянемо клас Activity і файли ресурсів, які визначають візуальний інтерфейс. Activity є класом, який по суті представляє окремий екран (сторінку) додатків або їхніх візуальний інтерфейс. Окремі activity, які вже безпосередньо використовуються в додатку, є спадкоємцями цього класу. Додаток може мати одну activity, а може і кілька. Кожна окрема activity задає окреме вікно для відображення.

Графічний інтерфейс користувача ґрунтується на ієрархії об'єктів View і ViewGroup. Об'єкти View представляють віджети, такі як кнопки або текстові поля. Об'єкти ViewGroup є контейнерами для віджетів, керують їхнім розташуванням і компонованням.

3.3 Опис програмного використання платформи Firebase

Firebase – це платформа для розробки мобільних додатків. Основними функціями застосування є [8-9]:

- а) хмарний Firestore, що дозволяє зберігати та синхронізувати дані додатків у глобальному масштабі;
- б) хмарні функції, які можуть запускати серверний код без керування сервером;
- в) автентифікація;
- г) хостинг;
- д) хмарне зберігання файлів;
- е) база даних в реальному часі.

Для створення системи були використані функції автентифікації, хмарного Firestore та хмарних функцій.

Використання хмарних технологій дає змогу відображати дані в режимі реального часу, завдяки тому що Firestore працює з шаблоном реактивного програмування.

У Firestore для зберігання даних використовуються колекції і документи. Документ - це запис, який містить будь-які поля. Документи об'єднуються в колекції. Також документ може містити вкладені колекції. Якщо проводити аналогію з SQL-

базою, то колекція - це таблиця, а документ - це запис в цій таблиці. Одна колекція може містити документи з різним набором полів (рисунок 3.15).

У Cloud Firestore, можна використовувати запити для отримання одного документа, визначеного документа, або всіх документів в колекції, які відповідають параметрам запити. Запити можуть містити ланцюжки з декількох фільтрів або фільтри з сортуванням. Також всі колекції індексуються за замовчуванням, з цього продуктивність запити прямо пропорційна результату, а не розміром колекції.

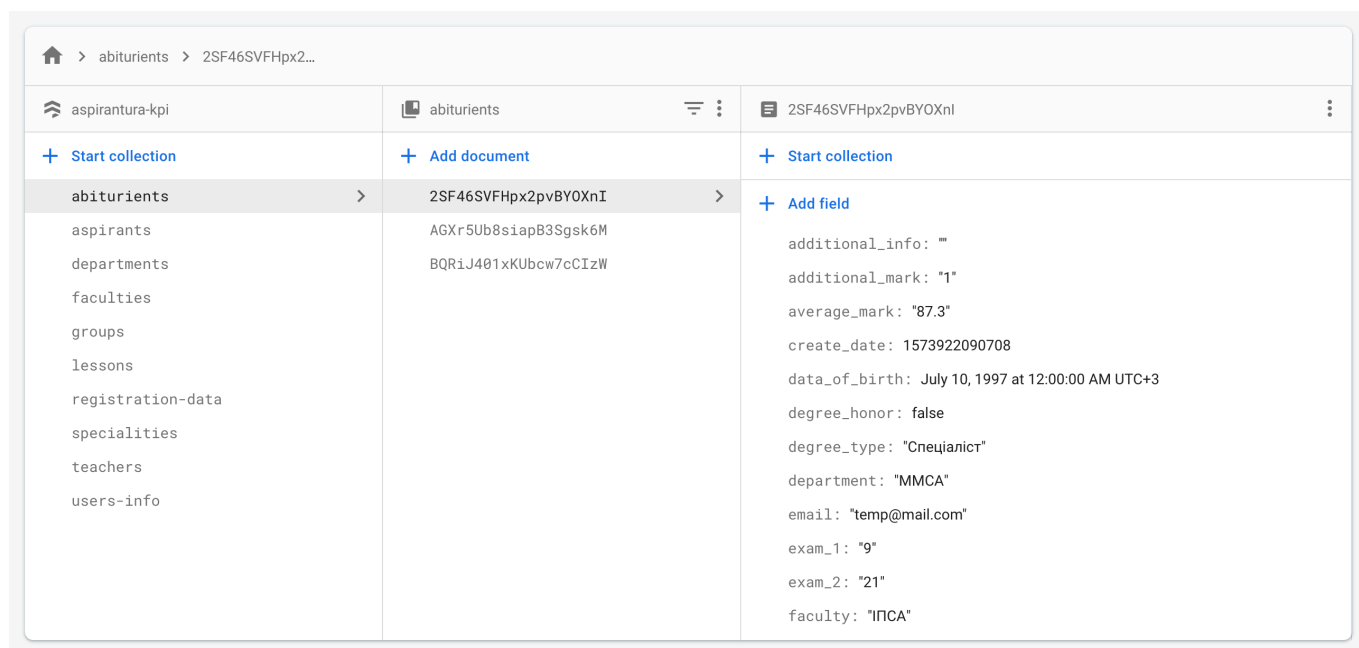


Рисунок 3.15 Структура бази даних системи

3.3.1 Створення документів колекції

Щоб мати можливість записувати в базу даних Firestore, спочатку потрібно отримати посилання на неї.

Потім потрібно або створити нову колекцію, або отримати посилання на існуючу колекцію, викликавши метод `collection()`.

Коли посилання на колекцію отримано, можна почати додавати до неї документи, викликавши метод `add()`.

Метод `add()` автоматично генерує і привласнює унікальний буквено-цифровий ідентифікатор кожному документу, який він створює. Якщо потрібно, щоб у документах були свої власні користувацькі ідентифікатори, спочатку потрібно вручну

створити ці документи, викликавши метод `document()`, який в свою чергу приймає унікальний рядок ідентифікатора. Потім можна заповнити документи, викликавши метод `set()`. Якщо унікальний ідентифікатор уже є в базі даних, то метод `set()` перезапише існуючий документ з цим ідентифікатором.

3.3.2 Створення підколекцій

Підтримка підколекції - одна з найбільш потужних функцій Firestore. Використовуючи підколекції, можна не тільки легко додавати вкладені структури в дані, але також бути впевненими, що запити будуть споживати мінімальні ресурси.

Створення підколекції дуже схоже на створення колекції. Все, що потрібно зробити, це викликати метод `collection()` на об'єкті `DocumentReference` і передати йому рядок, який буде використовуватися як ім'я підколекції.

3.3.3 Виконання запитів

Виконати операцію читання у базі даних Firestore дуже просто, якщо відомий ідентифікатор документа, який потрібно прочитати. Щоб безпосередньо отримати посилання на документ, потрібно викликати методи `collection()` і `document()`.

Якщо ідентифікатор документа невідомий, потрібно буде запустити традиційний запит для всієї колекції. API Firestore надає інтуїтивно названі методи фільтрації, такі як `whereEqualTo()`, `whereLessThan()` і `whereGreaterThan()`. Оскільки методи фільтрації можуть повертати кілька документів в якості їх результатів, знадобиться цикл всередині функції обробки результатів.

3.3.4 Видалення даних

Щоб видалити документ з відомим ідентифікатором, все, що потрібно зробити, це отримати посилання на нього, а потім викликати метод `delete()`.

3.4 Висновки до розділу

У цьому розділі наведено опис програмного забезпечення, що дає змогу брати участь у вступній компанії абітурієнтам та адмініструвати процес вступу.

Були створені модульні веб та Android додатки, які містить в собі компоненти та сервіси для роботи з базою даних Firestore та які об'єднують в собі шаблони програмування, такі як Dependency Injection.

Програмне забезпечення дозволяє супроводжувати систему вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації до вищого навчального закладу.

4 ІНТЕРФЕЙС ТА РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

Розроблена система є доступною для будь-якого користувача, який має встановлений сучасний браузер на своєму пристрої або смартфон на операційній системі Android. Для запуску системи потрібно перейти за посиланням, завантажити веб-додаток на локальному сервері або встановити додаток на смартфон.

4.1 Інтерфейс системи

На головній сторінці системи (рисунок 4.1, рисунок 4.2) користувач в ролі абітурієнта може подати реєстраційну форму або перевірити рейтинговий бал. Адміністратор може авторизуватись в системі для подальшої роботи з даними вступників.

Аспірантура КПІ

Для абітурієнтів

Подати реєстраційну форму

Переглянути рейтинг

Вхід

Email

wiggledima@gmail.com

Пароль

.....

Увійти

Рисунок 4.1 Головна сторінка системи

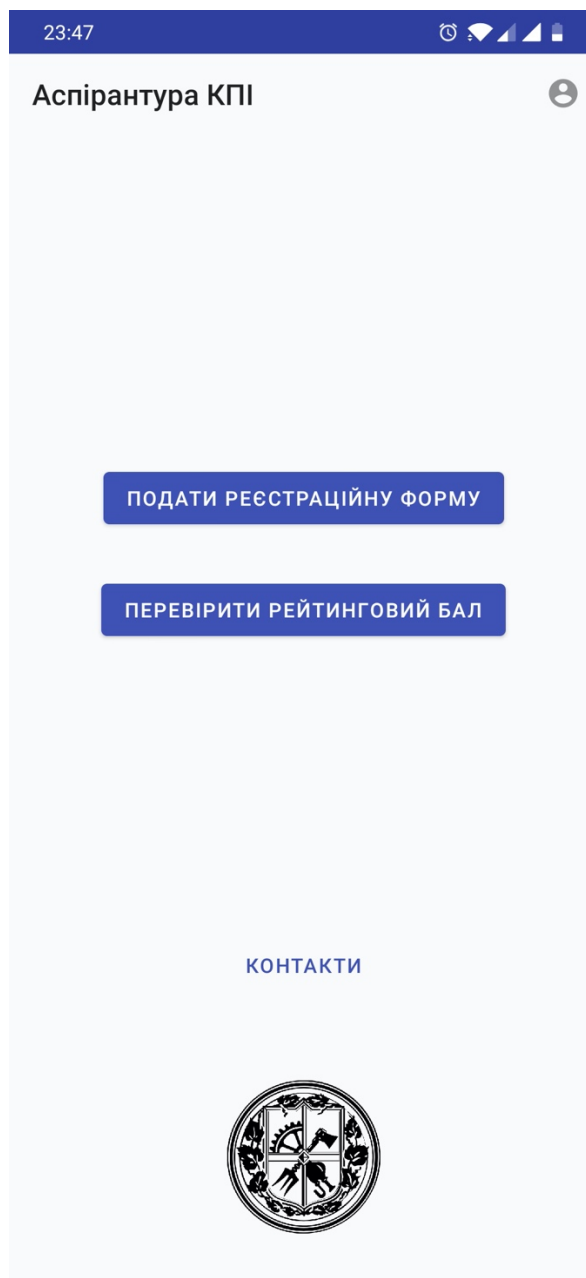


Рисунок 4.2 Головна сторінка системи на смартфоні

4.1.1 Функціональні можливості абітурієнта

Сторінка подачі реєстраційної форми складається з трьох етапів.

На першому етапі абітурієнт вводить свої дані (рисунок 4.3) :

- а) прізвище, ім'я, по батькові;
- б) стать;
- в) дата народження.

Аспірантура КПІ

Реєстраційна форма абітурієнта

1 Введіть своє ПІБ

Прізвище: Вradій

Ім'я: Дмитро

По батькові: Вікторович

Стать: Чоловіча

09.03.1997

Далі

2 Оберіть спеціальність, факультет і кафедру

3 Вкажіть додаткову інформацію

Рисунок 4.3 Перший етап подачі реєстраційної форми

23:48

До головного меню

Введіть своє ПІБ

Введіть прізвище

Введіть ім'я

Введіть по батькові

Стать

Дата народження

Продовжити реєстрацію

Рисунок 4.4 Перший етап подачі реєстраційної форми на смартфоні

На другому етапі абітурієнт обирає спеціальність, факультет та кафедру (рисунок 4.5, рисунок 4.6).

Аспірантура КПІ

Реєстраційна форма абітурієнта

1 Введіть своє ПІБ

2 **Оберіть спеціальність, факультет і кафедру**

Спеціальність
122 - Комп'ютерні н...

Факультет
ТЕФ

Кафедра
АПЕПС; Кафедра ав...

Назад Далі

3 Вкажіть додаткову інформацію

Рисунок 4.5 Другий етап подачі реєстраційної форми

23:49

До сторінки ПІБ

Оберіть спеціальність, факультет і кафедру

Спеціальність
122 - Комп'ютерні науки

Факультет
ТЕФ

Кафедра
ання енергетичних процесів та систем

Продовжити реєстрацію

Рисунок 4.6 Другий етап подачі реєстраційної форми на смартфоні

Третій етап подачі реєстраційної форми супроводжується заповненням даних полів (рисунок 4.7, рисунок 4.8):

- а) форма навчання;
- б) форма оплати;
- в) іноземна мова;
- г) ВНЗ який закінчив абітурієнт;
- д) рік випуску;
- е) освітній ступінь;
- ж) диплом з відзнакою;
- з) електронна пошта;
- и) телефон;
- к) чи потребує місця в гуртожитку;
- л) середній бал диплому за 100-бальною шкалою;
- м) кількість публікацій;

The screenshot shows the 'Аспірантура КПІ' (Aspirantura KPI) registration form, step 3: 'Вкажіть додаткову інформацію' (Provide additional information). The form includes the following fields and options:

- Форма навчання** (Form of study): Денна (Daytime)
- Форма оплати** (Form of payment): Держзамовлення (State order)
- Іноземна мова** (Foreign language): Англійська (English)
- Який ВНЗ закінчив** (Which university did you graduate from): КПІ (KPI)
- Рік випуску** (Year of graduation): 2019
- Спеціаліст/Магістр** (Specialist/Master): Магістр (Master)
- ☐ З відзнакою (With distinction)
- Email**: dima.vrady@gmail.com
- Телефон** (Phone): +380904159090
- ☒ **Потребую місце проживання в гуртожитку на період навчання** (Need a place of residence in a dormitory for the period of study)
- Середній бал диплому за 100-...** (Average diploma score out of 100...): 85
- Кількість публікацій** (Number of publications): 0
- Прізвище передбачуваного наукового керівника** (Surname of the prospective scientific supervisor): Коваль О.В. (Koval O.V.)
- Особливі відзнаки** (Special achievements (wins in olympiads, patents, etc.)): (empty field)
- Додаткова інформація** (Additional information): (empty field)

At the bottom, there are two buttons: 'Назад' (Back) and 'Завершити реєстрацію' (Complete registration).

Рисунок 4.7 Третій етап подачі реєстраційної форми

- н) прізвище передбачуваного наукового керівника;
- о) особливі відзнаки (перемоги на олімпіадах, патенти, тощо);
- п) додаткова інформація.

23:49 ← До сторінки спеціальності

Вкажіть додаткову інформацію

Форма навчання

Денна

Вечірня

Заочна

Який ВНЗ закінчив

0 / 50

Рік випуску

Спеціаліст/Магістр

☐ З відзнакою

Email

Телефон

☐ Потребую місце проживання в гуртожитку на період навчання

23:50 ← До сторінки спеціальності

☐ З відзнакою

Email

Телефон

☐ Потребую місце проживання в гуртожитку на період навчання

Середній бал диплому

Кількість публікацій

0

ПІБ передбачуваного наукового кері...

Особливі відзнаки (перемоги на олі...

0 / 200

Додаткова інформація

0 / 200

Завершити реєстрацію

Рисунок 4.8 Третій етап подачі реєстраційної форми на смартфоні

Після вдалої реєстрації, абітурієнт має змогу перевірити свій рейтинговий бал за допомогою форми пошуку абітурієнтів (рисунок 4.9).

ПІБ	Факультет	Спеціальність	Рейтинговий бал
Матвієнко Андрій Миколайович	ІПСА	111	15.315000000000001

Рисунок 4.9 Форма пошуку абітурієнтів для перевірки рейтингового балу

Введіть ПІБ

Матвієнко

ПОШУК

ПІБ: Матвієнко Андрій Миколайович

Факультет: ІПСА

Спеціальність: 111




Рейтинговий бал: 15.315000000000001




Рисунок 4.10 Форма пошуку абітурієнтів для перевірки рейтингового балу на смартфоні

Абітурієнт має змогу переглянути список вступників за спеціальністю та визначити своє місце в рейтингу (рисунки 4.11, 4.12).

<div>  Аспірантура КПІ </div>					
Рейтинговий список вступників за спеціальністю 111 (Математика). Форма навчання: Денна. Кількість бюджетних місць: 3					
№	ПІБ	Факультет	Спеціальність	Форма навчання	Рейтинговий бал
1	Матвієнко Андрій Миколайович	ІПСА	111	Денна	15.315000000000001
2	Магула Стас Григорович	ІПСА	111	Денна	13.93
3	Малиновський Андрій Михайлович	ІПСА	111	Денна	13.804999999999998
4	Пригожина Анастасія Юріївна	ІПСА	111	Денна	13.07

Рисунок 4.11 Рейтинговий список вступників

23:54   

← До сторінки пошуку

Рейтингова таблиця вступників за спеціальністю 111(Математика). Форма навчання - денна.
Обсяг прийому: 3

1	Матвієнко Андрій Миколайович	Рейтинговий бал: 15.32
2	Магула Стас Григорович	Рейтинговий бал: 13.93
3	Малиновський Андрій Михайлович	Рейтинговий бал: 13.81
4	Пригожина Анастасія Юріївна	Рейтинговий бал: 13.07

Рисунок 4.12 Рейтинговий список вступників на смартфоні

За допомогою мобільного додатку користувач може переглянути контакти та розташування відділу аспірантури (рисунк 4.13).

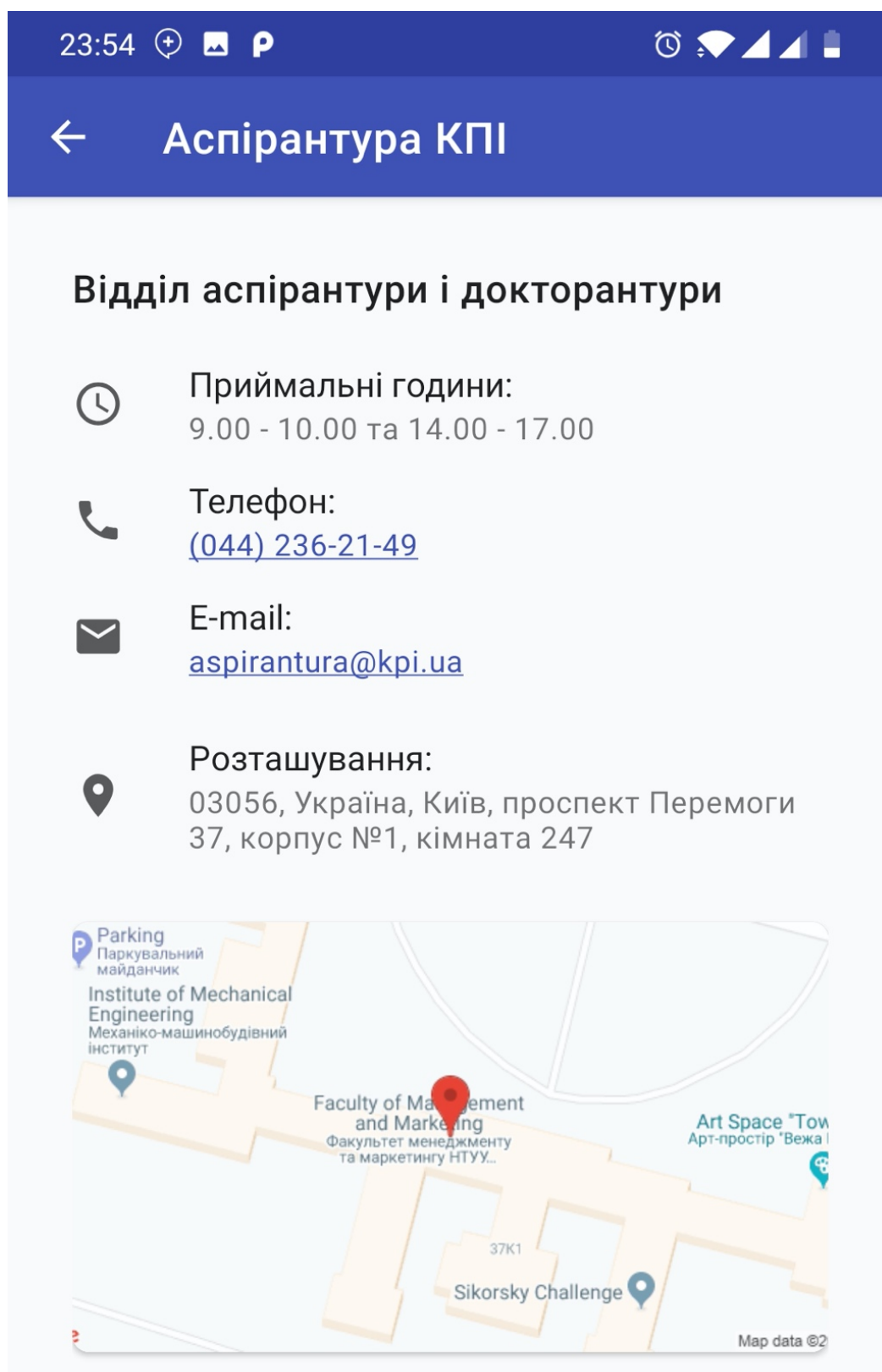


Рисунок 4.13 Сторінка контактів відділа аспірантури

4.1.2 Функціональні можливості адміністратора

Після успішної авторизації адміністратор системи може переглянути список поданих заяв (рисунок 4.14). В таблиці відображається перелік коротких даних вступників та кнопка, яка веде до детальної сторінки поданої заяви.

Аспірантура КПІ

Вітаємо, Дмитро Врадій

Подані заяви

Абітурієнти

Зараховані

Статистика

Пошук

ПІБ	Факультет	Спеціальність	
Северин Микита Андрійович	ФММ	75	Детальніше
Шишута Анатолій Миколайович	ТЕФ	144	Детальніше
Козловський Андрій Геннадійович	ММІ	131	Детальніше
Борщик Сергій Олександрович	ММІ	133	Детальніше
Солончук Ірина Вікторівна	ФСП	81	Детальніше
Котковський Богдан Сергійович	ФБМІ	163	Детальніше
Романенко Ірина Ігорівна	ТЕФ	143	Детальніше
Щоголева Світлана Геннадіївна	ММІ	173	Детальніше
Мощенко Ілля Михайлович	ХТФ	161	Детальніше
Назаренко Олена Володимирівна	ВПІ	186	Детальніше
Гумінський Артем Миколайович	ТЕФ	122	Детальніше
Бойко Дмитро Сергійович	ТЕФ	122	Детальніше
Євтушенко Артем Михайлович	ТЕФ	121	Детальніше

localhost:4200/cabinet/registration-data

КПІ 2019

Рисунок 4.14 Список поданих заяв

На детальній сторінці поданої заяви (рисунок 4.15), адміністратор може переглянути дані вступника та прийняти рішення підтвердити чи відмовити в реєстрації абітурієнту.

Якщо адміністратор відмовляє в реєстрації, реєстраційна картка абітурієнта видаляється з системи. Після підтвердження реєстрації картка вступника переходить до списку абітурієнтів.

Абітурієнт:	Северин Микита Андрійович
Стать:	Чоловіча
Дата народження:	15 лип. 1994 р.
Факультет:	ФММ
Кафедра:	ПМФММ
Спеціальність:	75
Форма навчання:	Вечірня
Форма оплати:	Держзамовлення
Іноземна мова:	Англійська
ВНЗ:	КПІ
Рік випуску:	2018
Диплом:	Магістр з відзнакою

Рисунок 4.15 Детальна сторінка поданої заяви вступника

На сторінці абітурієнтів адміністратору доступна до перегляду таблиця вступників, реєстрація яких була підтверджена (рисунок 4.16).

ПІБ	Факультет	Спеціальність	
Матвієнко Андрій Миколайович	ІПСА	111	Детальніше
Магула Стас Григорович	ІПСА	111	Детальніше
Пригожина Анастасія Юрівна	ІПСА	111	Детальніше
Малиновський Андрій Михайлович	ІПСА	111	Детальніше

Рисунок 4.16 Список абітурієнтів, реєстрація яких підтверджена

На детальній сторінці (рисунок 4.17), адміністратор може переглянути дані абітурієнта, а також занести бали за вступні екзамени до системи. Система автоматично згенерує рейтинговий бал та розмістить у рейтинговому списку вступників.

Аспірантура КПІ

Вітаємо, Дмитро Врадій

Подані заяви

Абітурієнти

Зараховані

Статистика

Абітурієнт: Матвієнко Андрій Миколайович

Екзамен з іноземної мови

9

Фаховий екзамен

21

Додаткові бали

1

☐ Потребує додаткового іспиту

Додатковий вступний іспит

Рейтинговий бал: 15.315000000000001

Зберегти зміни

Зарахувати абітурієнта

Видалити з бази

Стать:	Чоловіча
Дата народження:	10 лип. 1997 р.
Факультет:	ІПСА
Кафедра:	ММСА
Спеціальність:	111

КПІ 2019

Рисунок 4.17 Детальна сторінка картки абітурієнта

На сторінці «Зараховані» відображаються всі зараховані (рисунок 4.18), адміністратор має змогу додати аспіранта до групи (рисунок 4.19).

Аспірантура КПІ

Вітаємо, Дмитро Врадій

Подані заяви

Абітурієнти

Зараховані

Статистика

Пошук

ПІБ	Факультет	Кафедра	Спеціальність	Група
Іванов Іван Іванович	ТЕФ	АПЕПС	122	тр-91ф
Мельник Ігор Валентинович	ТЕФ	АПЕПС	122	тв-91ф
Некрашевич Олена Василівна	ТЕФ	АТЕП	151	та-91ф
Шевченко Іван Олексійович	ТЕФ	АПЕПС	121	тр-91ф

Рисунок 4.18 Список зарахованих

Аспірантура КПІ

Вітаємо, Дмитро Врадій

Подані заяви

Абітурієнти

Зараховані

Статистика

Пошук

ПІБ	Факультет	Кафедра	Спеціальність	Група
Іванов Іван Іванович	ТЕФ	АПЕПС	122	тр-91ф
Коваленко Андрій Іванович	ТЕФ	АПЕПС	122	Додати в групу
Мельник Ігор Валентинович	ТЕФ	АПЕПС	122	тв-91ф
Малиновський Андрій Мих		ММСА	111	Додати в групу
Некрашевич Олена Василів		АТЕП	151	та-91ф
Шевченко Іван Олексійович		АПЕПС	121	тр-91ф

Виберіть групу

тв-91ф

тр-91ф

Рисунок 4.19 Додавання зарахованого абітурієнта до групи

Адміністратору системи доступна до перегляду статистика вступної компанії по різним критеріям (рисунок 4.20):

- а) факультет – подана заяві
- б) факультет – абітурієнти
- в) факультет - зараховані

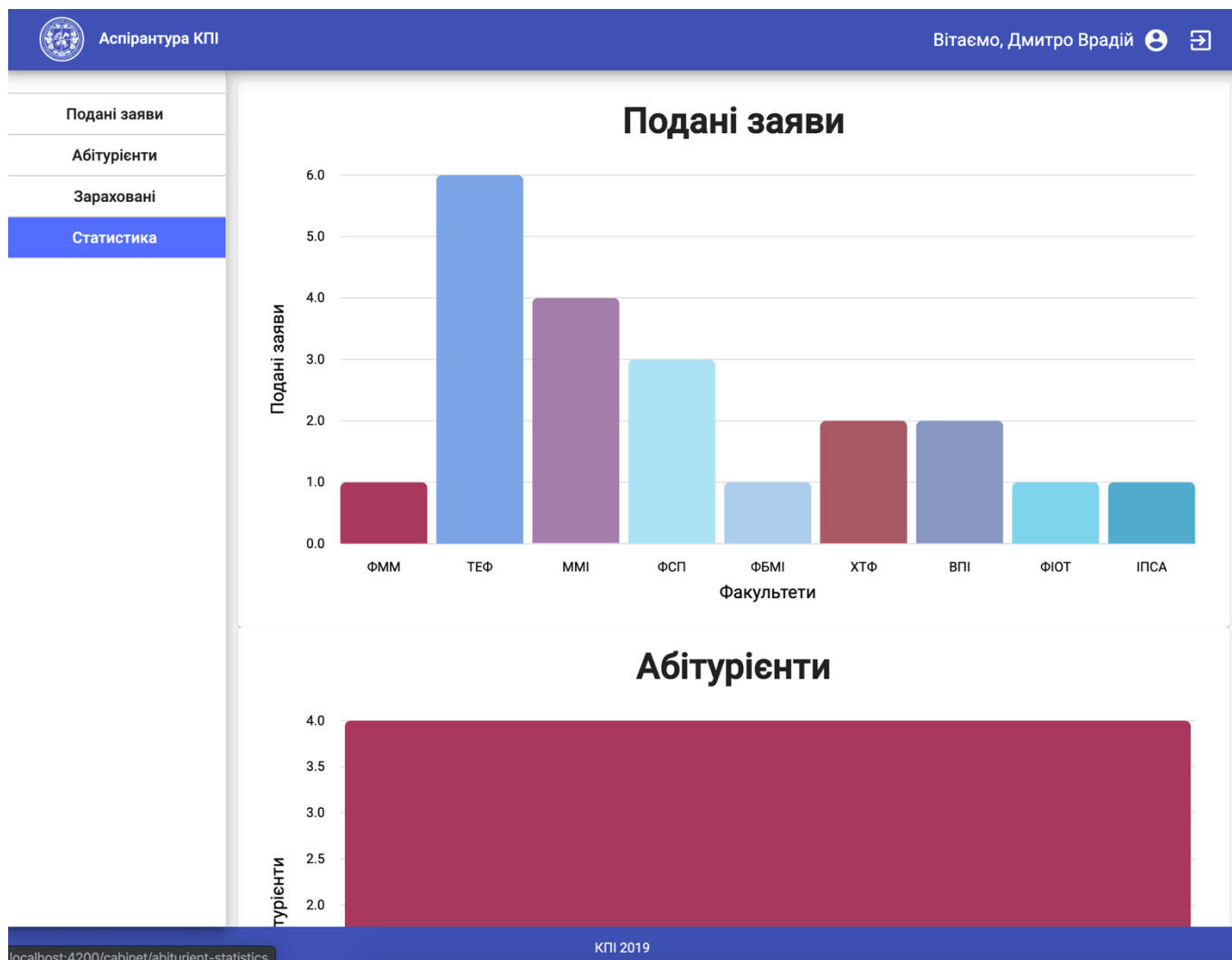


Рисунок 4.20 Статистика вступної компанії

4.2 Висновки до розділу

У цьому розділі наведено опис програмного забезпечення, що дає змогу проводити вступну компанію до аспірантури у вищому навчальному закладі.

Програмне забезпечення дозволяє абітурієнту подати реєстраційну форму та брати участь у вступній компанії. Користувач може переглядати свій рейтинговий бал в режимі реального часу.

Адміністратор системи забезпечений функціоналом, щоб проводити всі етапи вступної компанії. Користувач переглядає реєстраційні форми та додає їх до бази системи. Для підтверджених абітурієнтів відповідальний вносить бали за вступні экзамены, а також зараховує їх до вищого навчального закладу і присвоює групу.

Розділ статистики дозволяє адміністратору проаналізувати проведення вступної компанії до аспірантури.

У ході тестування додатку не було помічено порушень в роботі, тому додаток можна використовувати на пристроях, які підтримують роботу з веб-системами та на смартфонах з ОС Android.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

В розділі описується ідея проекту та визначаються загальні напрями використання продукту, а також його відмінність від конкурентів, аналізуються ринкові можливості щодо його реалізації. На базі аналізу ринкового середовища розроблена стратегія ринкового впровадження продукту в межах проекту [21].

5.1 Опис ідеї проекту

Сьогодні ринок програмних систем супроводження вступу до вищих навчальних закладів представлений не великою кількістю пропозицій. Існує державний програмний комплекс вступу до вищих навчальних закладів ІС «Конкурс».

Особливостями інформаційної системи «Конкурс» є:

- а) web-інтерфейс, який відображає процес вступної компанії;
- б) зручний пошук вступників за різними критеріями;
- в) автоматизований підрахунок конкурсної позиції;
- г) доступність системи з різних платформ;
- д) перегляд статистики;
- е) розширюваність системи;
- ж) державна підтримка;
- з) наявність API для використання іншими додатками.

Проаналізувавши систему, можна сказати, що інформаційна система «Конкурс» дає змогу проводити вступну компанію для вступників на перший та другий освітній рівень, але вона не забезпечує процес вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації. Зважаючи на це, є потреба у використанні системи, яка дозволить вирішити проблему для вищого навчального закладу (таблиця 5.1).

Таблиця 5.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення системи для супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації.	1. Адміністрація ВНЗ, яка проводить вступну компанію.	Автоматизація процесів адміністрування вступної компанії, швидкий доступ до бази даних вступників, можливість підключення системи до інших програмних модулів ВНЗ.
	2. Абітурієнти, які приймають участь в вступній компанії.	Доступність та відкритість подачі документів до вступної комісії, перевірка рейтингового балу та положення в рейтинговому списку.

Аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів наведено в таблиці 5.2.

Таблиця 5.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів		W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	ІС «Конкурс»			
.	Економічні:	\$100	\$5000/рік,	- -	- -	вартість обслуго

	вартість обслуговування; вартість експлуатації; вартість утилізації; вартість витратних матеріалів; вартість ремонту.	входить у вартість обслуговування безкоштовно входить у вартість обслуговування входить у вартість обслуговування	входить у вартість обслуговування безкоштовно входить у вартість обслуговування входить у вартість обслуговування	- - - -	вартість утилізації така ж як у конкурентів вартість витратних матеріалів така ж як у конкурентів вартість ремонту така ж як у конкурентів -	врування дешевша на ніж у конкурентів - - - присутні більш вигідні пропозиції знижки
.	Призначення (технічні) ОС;	Microsoft Windows 10, Ubuntu, Linux MacOS, Android OS	Microsoft Windows 10, Ubuntu, Linux MacOS	-	Для нормального функціонування ПЗ необхідне технічне	Додатково є підтримка ОС Android, тому існує можливість

					обладна ння на популяр них операцій них система х таких як Window s, Linus, MacOS, Ubuntu. Це дає змогу запустит и програм ний продукт майже на всіх персона льних комп'ют ерах	запуску додатку на мобільн их телефо нах
.	Захищеність і прозорість	Програмний продукт має захищену	Програмний продукт дає змогу	-	Програм не забезпеч	-

		адміністративн у частину у вигляді автентифікації. Прозорість досягається доступ до відкритих даних	переглядати відкриті дані вступників, шукати рейтингові списки за різними критеріями. Захищеність досягається тим, що використовуєть ся відкрите API		ення розробл ене за допомог ою сучасни х техноло гій, тому має всі причини бути серед програм них комплек сів даного рівня.	
.	Використані технології: Angular Authentication Firebase Firestore Cloud functions	+	-	-	Деякі техноло гії викорис товують ся в конкуре нтів	Веб додаток розробл ений за допомо гою мови TypeScr ipt та

	Hosting Android					фреймворку Angular, що є сучасними програвними засобами реалізації веб-систем. Використання хмарної платформи Firebase дає змогу використовувати сервіси такі як Firebase Firestore, Hosting,
--	--------------------	--	--	--	--	---

						Cloud function Викори стання мови програ муванн я Kotlin дає змогу створит и Android додаток , що розшир ить коло потенці йних клієнтів
	Кросплатфор мність	Web додаток та мобільний додаток на основі ОС Android	Web додаток.		-	Наявніс ть Android додатку дає змогу збільши ти можлив

						ості при розробці маркетингового плану для залучення нових клієнтів та інвестицій. Web-додаток додає кросплатформність
	Дизайн	Web-інтерфейс оснований на сучасних методологія користувацького інтерфейсу та мобільний інтерфейс у	Web-інтерфейс, оснований на використанні бібліотеки стилізації веб сторінок.	-	Існують можливості аналізу інтерфейсу конкурентів, що надає	-

		стилях ОС Android.			змогу створит и унікальн ий користу вацький інтерфе йс, який дасть змогу залучит и нових клієнтів до користу вання програм ним продукт ом. Це дасть можливі сть створит и конкуре нтоздат ний продукт	
--	--	-----------------------	--	--	--	--

Результатом дослідження сильних, слабких та нейтральних характеристик проекту є:

а) проаналізовані техніко-економічні характеристики існуючих на ринку конкурентів, що дало змогу спрогнозувати частку майбутніх користувачів програмного комплексу;

б) потенційне програмне забезпечення є конкурентоспроможним, так як представляє собою продукт з функціональними можливостями, які задовольняють майбутніх користувачів;

в) потенційний проект має 40 відсотків нейтральної сторони та 60 відсотків сильної сторони, що свідчить про конкурентоспроможність продукту;

г) проаналізовані характеристики дають змогу вирішити, що потенційний проект є перспективним продуктом, так як є удосконаленням наявних систем та додаванням нового функціоналу.

5.2 Технологічний аудит ідеї проекту

Для вирішення проблеми розробки проекту проведений технологічний аудит ідеї проекту (таблиця 5.3).

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Надання чіткого зрозумілого інтерфейсу для ведення вступної компанії	HTML, CSS, JavaScript, TypeScript, Angular, Kotlin	Наявні	Доступні для вільного користування

2.	Прийом та обробка даних	Firebase Firestore	Наявні	Доступні для вільного користування
3.	Можливість завантаження системи до хмарного сервера	Firebase Hosting	Наявні	Доступні для вільного користування
5.	Механізм автентифікації та авторизації користувача.	Firebase Auth	Наявні	Доступні для вільного користування
Обрана технологія реалізації ідеї проекту: програмний продукт розробляється у середовищах Jet brains PhpStorm та Android Studio, використовуючи мови програмування TypeScript та Kotlin, хмарна платформа Firebase для роботи з базою даних та хостингом.				

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Проведений аналіз ринкових можливостей запуску стартап-проекту та створена попередня характеристика потенційного ринку стартап-проекту (таблиця 5.4).

Таблиця 5.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	1 головний гравець
2.	Загальний обсяг продаж, грн/ум.од	немає даних
3.	Динаміка ринку (якісна оцінка)	зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	затвердження продукту регуляторним органом

5.	Специфічні вимоги до стандартизації та сертифікації	відсутні
6.	Середня норма рентабельності в галузі (або по ринку), %	+20%

На основі розглянутих характеристик показників стану ринку, можна зробити висновок, що ринок є привабливим для входження за попереднім оцінюванням.

Проведений аналіз характеристик потенційних клієнтів стартап-проекту (таблиця 5.5).

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Створення системи для супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації	Вступники до аспірантури, приймальна комісія	ІТ компанії (розробники, адміністратори) — вступники, які будуть подавати заяви та переглядати рейтинг; — члени приймальної комісії, які будуть адмініструвати систему	Програмний комплекс повинен забезпечувати зручний інтерфейс для проведення вступної компанії. Додаток повинен забезпечувати доступність на різних платформах та

				мобільних пристроях Android. Система має забезпечувати прозорість доступу до даних
--	--	--	--	--

Аналіз характеристик клієнтів показав, що цільовою аудиторією є вступники та приймальна комісія університету.

Були проаналізовані фактори загроз стартап-проекту (таблиця 5.6).

Таблиця 5.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Бюджет	Ризик перевищення виділеного на проект бюджету	Цілі і завдання по розробці ПО можуть змінюватися в процесі його створення, і для того, щоб продукт залишався життєздатним, необхідно вміти підлаштовуватися під можливі зміни
2.	Працівники команди з малим досвідом	Працівникам з малим досвідом може знадобитись більше часу для виконання завдань. Це може змінити строки здачі проекту, оскільки для ознайомлення з новими технологіями, бізнес-процесами потребує часу.	Щоб мінімізувати цей фактор загрози, потрібно запланувати достатньо часу на навчання нових працівників.

3.	Час	При розробці програмного забезпечення дуже поширені затримки релізів продукції, що є результатом неправильного планування, вкрай стислі терміни і нездатності розробників адаптуватися до мінливих вимог по відношенню до продукту	Правильна організація процесу Процес повинен бути гнучким, щоб розробники могли швидко адаптуватися до мінливих вимог, мали можливість швидко надавати виправлений продукт замовнику, і могли точно визначати кількість часу, необхідне для виконання того чи іншого завдання.
----	-----	--	---

Були проаналізовані фактори можливостей (таблиця 5.7).

Таблиця 5.7. Фактори можливостей

№ п/п	Фактор	Зміст	Можлива реакція компанії
1.	Аналіз рішень конкурентів	Реалізація програмного продукту з урахуванням помилок конкурентів та введенням нового функціоналу, який не присутній в продуктах конкурентів.	Розробка дорожньої карти з реалізації проекту без виникнення помилок конкурентів.
2.	Розвиток технологій і введення програм, що дозволяють знизити витрати.	Реалізація нових технологій в проекті дозволить додати нові можливості для потенційних користувачів та спростити процес розробки.	Чітке розуміння напрямку проекту в технологічному аспекті та створення дорожньої карти.

		Це дасть змогу зменшити витрати на реалізацію проекту, так як час на виконання задач буде зменшений.	
3.	Можливості з розширення сфери впливу проекту	За рахунок розширення сфери впливу проекту, можна запропонувати систему іншим вищим навчальним закладам	Збільшення потенційних користувачів продукту.

Таблиця 5.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Чиста конкуренція	Галузь є конкурентною, не існує явних лідерів	Створення якісного продукту, успішна маркетингова компанія
2. Національний рівень конкурентної боротьби	Конкуренти в середині країни	Вивчення конкурентів та створення унікального продукту
3. Конкуренція за видами товарів: - товарно-видова	Характеристика проявляється в конкуренції ПЗ одного виду	Створення програмного забезпечення, яке буде задовольняти потреби клієнтів та зменшувати їх витрати на утримання такого програмного забезпечення

4. За характером конкурентних переваг – нецінова конкуренція	Конкуренція проявляється в різних функціональних можливостях продукту	Розширення та створення нових можливостей програмного продукту
5. За інтенсивністю – не марочна конкуренція	Для споживачів головне функціонал	Створення програмного продукту, який задовольняє клієнтів

Проведений ступеневий аналіз конкуренції на ринку показав, що в проекті є перспективи з виходом на ринок. Можливі дії компанії допоможуть вирішити особливості конкурентного середовища.

Таблиця 5.9. Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	ІС «Конкурс»	Система конкурентів успішно працює на ринку	Державні підприємства	Вищі навчальні заклади	немає
Висновки:	Конкурентна боротьба з невеликою інтенсивністю, так як конкуренти представляють функціонал, якого немає в	Створення нових функціональних можливостей програмної системи	Постачальники не диктують умови співпраці	Умови клієнтів регламентуються нормативно-правовими актами	немає

	потенційному продукті				
--	--------------------------	--	--	--	--

Аналіз конкуренції в галузі за М. Портером показав, що можливий вихід на ринок, бо конкурентна боротьба має невелику інтенсивність і прямі конкуренти не поставляють функціональні можливості задані потенційним програмним продуктом. Умови клієнтів регламентуються нормативно-правовими актами, а постачальники не диктують умови співпраці.

Таблиця 5.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Вимоги клієнтів	Вимоги клієнтів створюють потребу розробки проекту
2.	Результативність	Система задовольняє вимоги клієнтів
3.	Маркетинговий потенціал	Спосіб подачі програмного продукту клієнтам
4.	Ціна програмного продукту	Набагато менша ніж в конкурентів
5.	Технічне обслуговування	Підтримка наявних версій та випуску нових

Результатом обґрунтування факторів конкурентоспроможності є такі фактори:

- а) вимоги клієнтів;
- б) результативність;
- в) маркетинговий потенціал;
- г) технічне обслуговування.

Таблиця 5.11. Порівняльний аналіз сильних та слабких сторін програмного забезпечення

		Бали	Рейтинг програм-конкурентів
--	--	------	-----------------------------

№ п/п	Фактор конкурентоспроможності	1-20	у порівнянні з системою «Аспірантура КПІ»						
			-3	-2	-1	0	+1	+2	+3
1.	Вимоги клієнтів	10				+			
2.	Результативність	15					+		
3.	Маркетинговий потенціал	10					+		
4.	Ціна програмного продукту	5			+				
5.	Технічне обслуговування	15						+	

Порівняльний аналіз сильних та слабких сторін «Системи супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації» показує, що вимоги клієнтів, результативність, маркетинговий потенціал, технічне обслуговування, а також ціна програмного продукту є сильними факторами конкурентоспроможності у порівнянні з найближчими конкурентами.

Таблиця 5.12. SWOT-аналіз стартап-проекту

Сильні сторони: Інтуїтивно-зрозумілий інтерфейс; швидкодія програмного комплексу; кросплатформність; доступність; відкритість.	Слабкі сторони: Малий досвід використання нового програмного продукту; присутність недоробок на початковому етапі продаж.
Можливості: Вихід на національний ринок; додавання нових функціональних можливостей.	Загрози: Присутність недоробок на ранньому етапі, може спричинити відтік клієнтів.

SWOT-аналіз стартап-проекту вказав на сильні сторони: інтуїтивно-зрозумілий інтерфейс; швидкодія програмного комплексу; кросплатформність; доступність; відкритість.

Слабкими сторонами є: малий досвід використання нового програмного продукту; присутність недоробок на початковому етапі продаж.

Можливості представлені такими показниками: вихід на національний ринок; додавання нових функціональних можливостей.

Загрозою проекту є присутність недоробок на ранньому етапі, може спричинити відтік клієнтів.

Таблиця 5.13 Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Залучення державних інвестицій на розробку програмного продукту	50%	12 міс.

На основі SWOT-аналізу проекту була створена альтернатива ринкової поведінки стартап-проекту. В якості альтернативи було обрано залучення державних інвестицій на розробку програмного продукту. Ймовірність отримання ресурсів дорівнює 50 відсотків та строк реалізації 12 місяців.

5.4 Розроблення ринкової стратегії проекту

Проведений вибір цільових груп потенційних споживачів (таблиця 5.14).

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту) за рік	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Вищі навчальні заклади	Готові	500	Мала конкуренція	Легка

В якості цільової групи потенційних споживачів було обрано вищі навчальні заклади. Інтенсивність конкуренції в сегменті мала в Україні та простота входу в сегмент є достатньо легкою.

Визначена базова стратегія розвитку (таблиця 5.15).

Таблиця 5.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Проведення вступних компанії для інших освітніх ступенів	Ексклюзивний розподіл	Низька ціна, легка установка на базі цільової групи	Стратегія лідерства по витратах

Для обраної альтернативи розвитку проекту було обрано стратегію охоплення ринку на основі ексклюзивного розподілу. Базовою стратегією розвитку є стратегію лідерства по витратах.

Таблиця 5.16 Визначення базової стратегії конкурентної поведінки

№ п/ п	Чи є проект «першопрох ідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурент ної поведінки
1.	Проект не є першопрох ідцем	Компанія буде забирати існуючих споживачів у конкурентів і шукати нових	Основні характеристики товару будуть схожими до конкурента	Стратегія позиціону вання

При визначенні базової стратегії конкурентної поведінки даного проекту, який не є «першопрохідцем», було обрано стратегію позиціонування. Компанія демонструє відмінності продукту від конкурентів: корисність, переваги. Таким чином компанія створює позиціонування на особливостях, які важливі клієнту.

Таблиця 5.17 Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Вартість, зручність	Знизити ціни на продукт та створити систему, яка буде зручна для використання клієнтами.	Ціна, інтуїтивний інтерфейс системи, крос-платформність	Доступність, надійність, швидкість розгортання

При визначенні стратегії позиціонування були обрані вимоги до товару цільової аудиторії такі, як вартість та зручність. Обрано базову стратегію розвитку – Знизити ціни на продукт та створити систему, яка буде зручна для використання клієнтами. Були обрані асоціації, які формують комплексну позицію проекту – доступність, надійність, швидкість розгортання.

5.5 Розроблення маркетингової програми стартап-проекту

При визначенні ключових переваг концепції потенційного товару було проаналізовано такі потреби (таблиця 5.18):

- а) доступність;
- б) сприятливий до користувача інтерфейс;
- в) швидкодія.

Таблиця 5.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує ПЗ	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Доступність	Кросплатформність, існує веб-сайт та мобільний додаток	Наявність доступу з різних пристроїв
2.	Сприятливий до користувача інтерфейс	Швидке розуміння процесу роботи з системою	Застосування сучасних підходів до створення інтерфейсу
3.	Швидкодія	Для запуску програмного забезпечення не потрібно великих технічних затрат	Невисока завантаженість системи

Вигода яку пропонує ПЗ представляє собою кросплатформність, швидке розуміння процесу роботи з системою, використання не потужних пристроїв, а ключовими перевагами перед конкурентами є наявність доступу з різних пристроїв, застосування сучасних підходів до створення інтерфейсу та невисока завантаженість системи.

Була розроблена трирівнева маркетингова модель товару (таблиця 5.19), в якій уточнюється ідея продукту, його фізичні складові та особливості процесу його надання.

Таблиця 5.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
--------------	----------------------

I. Товар за задумом	Створити систему для супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Можливість подачі вступних заяв абітурієнтами	-	-
	Можливість перевірки рейтингового балу абітурієнтом		
	Можливість перевірки рейтингових списків абітурієнтів		
	Можливість адміністрування вступної компанії		
	Захищеність адміністративної частини		
Перегляд статистики вступної компанії			
Сучасний інтерфейс, який є зрозумілим для користувача			
	Якість: стандарти, нормативи, параметри тестування		
	Марка: «Аспірантура КПП»		
III. Товар із підкріпленням	До продажу: презентація програмного продукту		
	Після продажу: розширення новим функціоналом		
За рахунок чого потенційний товар буде захищено від копіювання: Ліцензування та закритий код			

До продажу товару потрібно провести презентацію програмного продукту, а після продажу потрібно розширювати новим функціоналом. Товар захищений від копіювання ліцензією та тим, що має закриту кодову базу.

Було проведено формування оптимальної системи збуту:

- вибрано специфіку закупівельної поведінки цільових клієнтів;
- визначено функції збуту, які має виконувати постачальник товару;
- вибрано глибину каналу збуту.

Таблиця 5.20. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Клієнти отримують програмне забезпечення від компанії-розробника	Налаштування програмного продукту на базі технічного обладнання замовника	Канал нульового рівня (виробник безпосередньо продає товар клієнту)	Через адміністрацію клієнта

Було обрано оптимальну систему збуту, при якій програмний продукт передається через адміністрацію клієнта. Постачальник має виконувати функції збуту: налаштовувати програмний продукт на базі технічного обладнання замовника.

5.6 Висновки до розділу

В результаті створення маркетингового аналізу стартап-проекту були визначені загальні напрями використання потенційного програмного продукту, а також відмінність від конкурентів.

Було визначено такі показники стартап-проекту:

- а) попит на даний програмний продукт існує, тому можна комерціалізувати проект;
- б) стан конкуренції та бар'єр входження не високий, тому перспективи впровадження продукту існують;
- в) подальша імплементація проекту є доцільною, так як продукт можна розширювати завдяки додаванню нових функцій.

ВИСНОВКИ

В результаті дослідження було розроблено систему супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації. Система була реалізована у вигляді веб-застосунку та Android додатку. Це сприяло покращенню навичок розробки програмного забезпечення за допомогою мов програмування TypeScript та Kotlin. Також були отримані навички роботи з хмарною платформою Firebase, яка містить в собі бази даних реального часу.

Завдяки використанню мови програмування TypeScript та фреймворку Angular 8 на боці веб-застосунку і мови програмування Kotlin з боку розробки на ОС Android, вдалося досягти поставлених цілей та забезпечити легку підтримку системи, можливість масштабування, підключення додаткових модулів та додавання нового функціоналу.

Використання клієнтського застосунку та мобільного додатку забезпечили дуже високу доступність програмного засобу та можливість запровадити єдиний інтерфейс для взаємодії з продуктом, незалежно від операційної системи, встановленої на користувацькому пристрої та його програмному забезпеченні.

Таким чином, розроблена система дає змогу брати участь у вступній компанії всім абітурієнтам та адмініструвати дані з боку приймальної комісії. Доступність забезпечує прозорий процес вступної компанії вищого навчального закладу. Адміністрування здійснюється в одному місці, що дає змогу пришвидшити та автоматизувати процеси, які раніше були паперовими.

Розроблена програмна система може використовуватись на великій кількості платформ, бо є кросплатформною. Для запуску системи потрібно використовувати сучасний браузер, який присутній на багатьох пристроях або мобільних телефонах під управлінням ОС Android.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

2. Міністерство освіти і науки України. Умови прийому на навчання до закладів вищої освіти України в 2019 році / Міністерство освіти і науки України — Київ, 2019 – 54 с.
3. Флэнаган Д. JavaScript. Подробное руководство / Дэвид Флэнаган — Символ-Плюс 2013. — 1080 с.
4. Стефанов С. JavaScript. Шаблоны проектирования / Стефанов Стоян. — Санкт-Петербург: Лори, 2011. — 272 с.
5. Хэмптон С. CSS3 для профессионалов / Сэм Хэмптон., 2016. — 183 с
6. Фримен А. Angular для профессионалов/ Адам Фримен., 2018. — 800 с
7. Файн Я. К. AngularJS. Принципы построения приложений / Яков Файн., 2014. — 208 с.
8. Файн Я., Моисеев А. Angular и TypeScript. Сайтостроение для профессионалов/ Яков Файн, Антон Моисеев., 2018. — 464 с
9. Uttam Agarwal. Hands-On Full Stack Development with Angular 5 and Firebase: Build real-time, serverless, and progressive web applications with Angular and Firebase / Uttam Agarwal, 2018 – 256 с.
10. Houssein Yahiaoui. Firebase Cookbook / Houssein Yahiaoui, 2017 p. – 288 с.
11. Руководство по Angular 6 [Електронний ресурс]. — Режим доступу до ресурсу: <https://metanit.com/kotlin/tutorial/>
12. Руководство по языку Kotlin [Електронний ресурс]. — Режим доступу до ресурсу: <https://metanit.com/web/angular2/>
13. Руководство по TypeScript [Електронний ресурс]. — Режим доступу до ресурсу: <https://metanit.com/web/typescript/>
14. ng-book: The Complete Book on Angular 6 [Електронний ресурс]. — Режим доступу до ресурсу: <https://www.ng-book.com/2/>
15. Angular - Architecture overview [Електронний ресурс]. — Режим доступу до ресурсу: <https://angular.io/guide/architecture>

16. Documentation – TypeScript [Електронний ресурс]. — Режим доступу до ресурсу: <https://www.typescriptlang.org/docs/home.html>
17. Бен Кристенсен — «Reactive Programming with RxJava: Creating Asynchronous», 2014 р. — 560 с.
18. Б. Харди, Б. Филлипс, К. Стюарт, К. Марсикано — «Андроид для профессионалов», 2014 р. — 475 с.
19. Портнякин И. — «Kotlin in Action», 2015 р. — 300 с.
20. Рето Майер — «Android 4», 2014 р. — 208 с.
21. Ashok Kumar S. Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase, 2018 р. – 394 с.
22. Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.

Додаток А

Система супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів
вищої кваліфікації

Апробації

УКР.НТУУ “КПІ ім. Ігоря Сікорського”.ТР4271_19М

Аркушів 5

2019

www.konferenciaonline.org.ua

**Міжнародна наукова
інтернет-конференція**

**Інформаційне суспільство:
технологічні, економічні
та технічні аспекти становлення**

(випуск 44)

ISSN 2522-932X

12 грудня 2019 р.

**Тернопіль
2019**

Зміст

Секція 1. Інформаційні системи і технології

Ваш Ю.В.

Дизайн та технології створення ігрового рушія..... 3

Верещак О.Р.

Прогнозування параметрів комп'ютерної мережі на основі інтелектуальних технологій5

Врадій Д.В.

Система супроводження та аналізу вступу спеціалістів науково-педагогічних кадрів вищої кваліфікації..... .8

Гумінський А.М.

Система супроводження та аналізу навчального процесу спеціалістів науково-педагогічних кадрів вищої кваліфікації..... 9

Гоголь В.М., Гуменюк Т.В.

Комп'ютерна система ідентифікації технологічних станів бурової установки 12

Дмитрієв В.С.

Комп'ютер та комп'ютерні системи..... 14

Мінько А.Ю.

Інформаційна безпека і засоби захисту інформації..... 16

Стелюк Б.Б., Олійник О.О.

Підходи до оцінки загроз інформаційної безпеки організації..... 17

Врадій Д.В., магістрант

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського», м. Київ.

Кафедра автоматизації проектування енергетичних процесів і систем

СИСТЕМА СУПРОВОДЖЕННЯ ТА АНАЛІЗУ ВСТУПУ СПЕЦІАЛІСТІВ НАУКОВО-ПЕДАГОГІЧНИХ КАДРІВ ВИЩОЇ КВАЛІФІКАЦІЇ

З року в рік постає проблема покращення якості процесу вступу до вищих навчальних закладів. Було розроблено автоматизовану систему, яка спрощує процес подання вступних заяв та перегляду змін в рейтингових списках. Дана система покриває перший та другий освітні рівні вищої освіти, тому проблема автоматизації процесу для цієї категорії вступників вирішена. Але існує ще третій освітній рівень вищої освіти, для якого не існує системи автоматизації процесу вступу. Це зумовлює актуальність розробки нових програмних засобів тому, що автоматизація процесу має на меті економію ресурсів, пришвидшення подачі та перевірки вступних заяв, недопустимість припущення помилок людського фактору.

Система може бути поділена на дві функціональні підсистеми:

- а) підсистема абітурієнта;
- б) підсистема адміністратора.

Абітурієнт має змогу подавати реєстраційні заяви та переглядати рейтингові списки вступників.

Адміністратор приймає рішення у реєстрації вступника в системі, а також супроводженням картки абітурієнта після проходження вступних екзаменів. Адміністратору доступна сторінка перегляду статистики вступної компанії. На рисунку 1 зображена діаграма прецедентів системи.

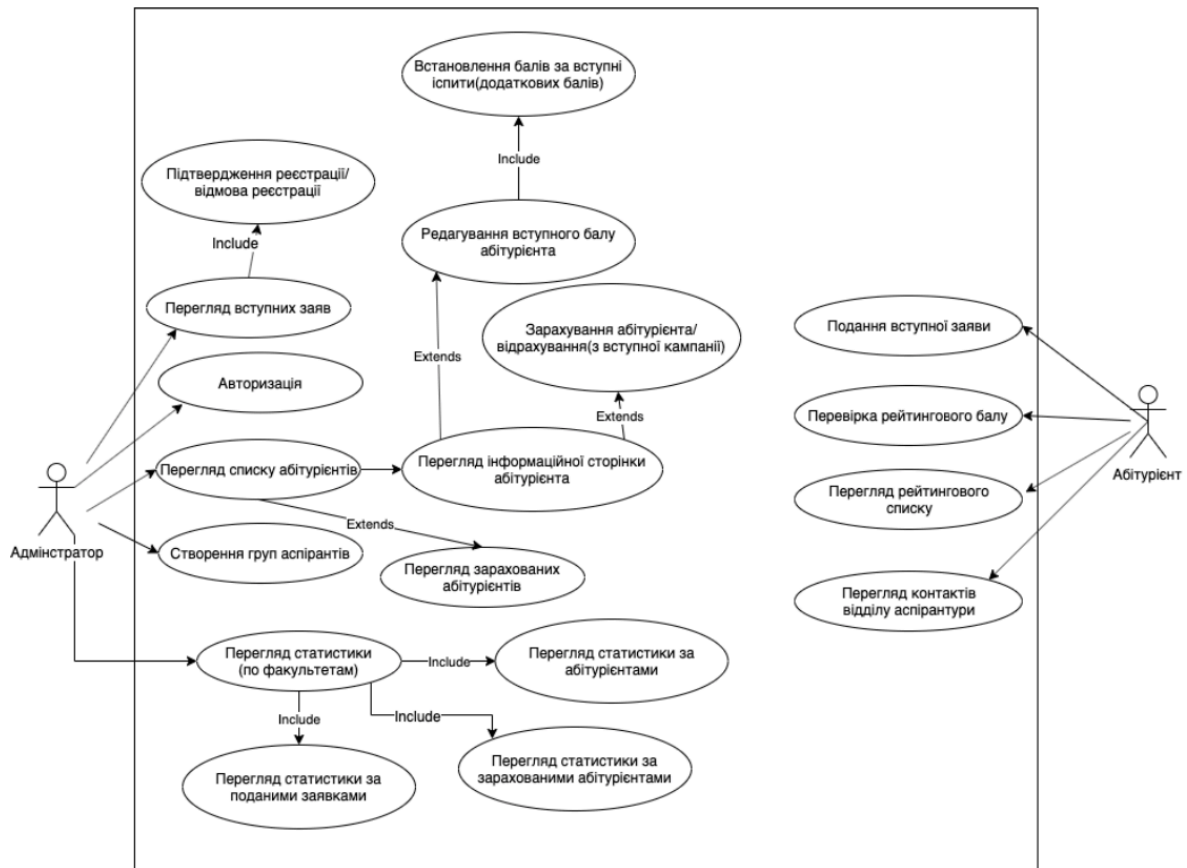


Рис. 1. Діаграма прецедентів системи

Під час розробки програмного продукту є правильний вибір засобів програмної реалізації та технологій. Це дає змогу вплинути на час розробки, якість та надійність.

Для реалізації веб-системи було використано мову програмування TypeScript та фреймворк Angular 8 [1], для мобільного додатку було використано мову програмування Kotlin [2]. Поєднання цих технологій дає змогу створити кросплатформний додаток. Це означає, що даний застосунок можна використовувати на великій кількості платформ: Android, Windows, Linux, MacOS.

Література

1. Руководство по Angular 8 [Електронний ресурс]. — Режим доступу до ресурсу: <https://metanit.com/web/angular2/>
2. Руководство по языку Kotlin [Електронний ресурс]. — Режим доступу до ресурсу: <https://metanit.com/kotlin/tutorial/>